



UNIVERSIDAD CARLOS III DE MADRID  
Ingeniería Informática

# Diseño e implementación de una aplicación Android para la gestión de pedidos en restaurantes

PROYECTO DE FIN DE CARRERA

Madrid, Febrero 2013

**Autora:** Carmen Fuentes Alba.  
**Tutores:** Alejandro Calderón Mateos.  
Javier Fernández Muñoz.



Título: Diseño e implementación de una aplicación Android para la gestión de pedidos en restaurantes.

Autora: Carmen Fuentes Alba.

Directores: Alejandro Calderón Mateos y Javier Fernández Muñoz.

## EL TRIBUNAL

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día \_\_\_\_ de \_\_\_\_\_ de 20\_\_ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE





*A mis padres, por su infinita paciencia.*

*A Pablo, por apoyarme y animarme a seguir adelante.*

*A mis tutores, Alejandro y Javier, por ser mis guías en el camino.*

*Qué hubiera sido de mí sin vosotros...*

*“Recorre el camino las veces que necesites, pero no intentes volver atrás.”*

**Super Mario Bros, 1985.**



# Resumen

Este proyecto describe el desarrollo de un sistema ideado para la gestión de pedidos en restaurantes a través de dispositivos Android. Se diseñarán distintos módulos para dar soporte a la funcionalidad que necesitará cada usuario. Los usuarios estarán en comunicación constante gracias a un servidor dedicado que permite las comunicaciones en remoto, quedando toda la información intercambiada registrada en base de datos. El presente documento tiene como finalidad la presentación del proceso completo llevado a cabo para la realización del proyecto.



# Abstract

The project carried out describes the development of an Android application able to manage the orders placed in a restaurant. Several modules will be designed to support the functionality needed for each user. Users will be in permanent communication thanks to a dedicated server that allows remote communications, keeping all information exchanged stored in a data base. The current document details the complete process needed to create the project.



# Índice de contenidos

<b>Capítulo 1. Introducción y Objetivos .....</b>	<b>21</b>
1.1    Visión general .....	22
1.2    Motivación .....	23
1.3    Objetivos .....	24
1.4    Fases de desarrollo.....	25
1.5    Estructura del documento.....	26
<b>Capítulo 2. Estado de la cuestión .....</b>	<b>27</b>
2.1    Evolución de la tecnología.....	28
2.1.1    Evolución de la tecnología en la hostelería .....	28
2.1.2    Evolución de la tecnología de las comunicaciones .....	30
2.2    Análisis de la plataforma móvil .....	32
2.2.1    Criterios de selección de plataforma móvil.....	34
2.2.2    iOS .....	37
2.2.3    Android.....	38
2.2.4    Plataforma móvil seleccionada .....	39
2.3    Análisis de aplicaciones .....	40

2.3.1	Comparativa .....	43
2.4	Tecnología utilizada.....	44
2.4.1	Herramientas básicas .....	44
2.4.2	Herramientas específicas .....	46
<b>Capítulo 3. Análisis, Diseño e Implementación.....</b>		<b>51</b>
3.1	Análisis.....	52
3.1.1	Requisitos de Usuario.....	52
3.1.2	Casos de Uso .....	60
3.1.3	Requisitos Software.....	72
3.1.4	Matriz de trazabilidad .....	88
3.2	Diseño.....	90
3.2.1	Diseño de la Arquitectura.....	90
3.2.2	Diseño de la Base de Datos .....	98
3.2.3	Diseño de Interfaz .....	105
3.2.4	Diagrama de navegación .....	126
3.3	Implementación .....	128
3.3.1	Servidor .....	128
3.3.2	Aplicación cliente y manager .....	136
3.4	Pruebas.....	141
3.4.1	Pruebas de funcionalidad.....	141
3.4.2	Pruebas de estrés .....	147
3.4.3	Matriz de trazabilidad de funcionalidad .....	149
<b>Capítulo 4. Planificación y Presupuesto .....</b>		<b>151</b>
4.1	Planificación .....	152
4.1.1	Diagrama de Gantt .....	153
4.2	Presupuesto.....	154
4.2.1	Resumen de tiempo dedicado .....	154
4.2.2	Desglose: Coste de personal .....	155
4.2.3	Desglose: Coste de Hardware .....	155
4.2.4	Desglose: Coste de Software.....	156
4.2.5	Resumen de costes.....	157
<b>Capítulo 5. Conclusiones y Líneas Futuras .....</b>		<b>159</b>
5.1	Conclusiones.....	160
5.1.1	Conclusiones del Proyecto .....	160



5.1.2	Conclusiones Personales .....	162
5.2	Líneas Futuras .....	163
<b>Glosario.....</b>		<b>169</b>
<b>Anexo A. Manual de Usuario Manager .....</b>		<b>173</b>
<b>Anexo B. Manual de Usuario Cliente.....</b>		<b>179</b>
<b>Anexo C. Implantación.....</b>		<b>187</b>
<b>Bibliografía .....</b>		<b>193</b>



# Índice de figuras

Figura 1 – Caja registradora con TPV e impresora de tickets.....	28
Figura 2 – Dispositivo PDA para realizar pedidos.....	29
Figura 3 – Dispositivos móviles. De izquierda a derecha: PDA, GPS, Smartphone, Tablet, MP3, videoconsola. ....	30
Figura 4 – Crecimiento del número de usuarios de dispositivos móviles. ....	31
Figura 5 – Diferentes dispositivos Tablet. De izquierda a derecha y adelante a atrás: Apple iPad 2 (negro y blanco), BlackBerry PlayBook, Samsung Galaxy tab 10.1 y Samsung Galaxy tab 8.9	36
Figura 6 – Crecimiento del número de aplicaciones para Android y iOS desde 2009 a 2012.....	39
Figura 7 – Interfaz de OrderingApps .....	40
Figura 8 – Interfaz de Snapfinger .....	41
Figura 9 – Interfaz de Food2U.....	42
Figura 10 – Interfaz de QlikBar .....	42
Figura 11 – Casos de Uso Cliente. Interfaz: carta del restaurante. ....	61
Figura 12 – Casos de Uso Cliente. Interfaz: pedido en curso (sin enviar a cocina). ....	64
Figura 13 – Casos de Uso Cliente. Interfaz: pedido realizado (enviado a cocina).....	66
Figura 14 – Casos de Uso Manager. Interfaz: pedidos activos.....	68
Figura 15 – Casos de Uso Manager. Interfaz: historial de pedidos.....	71
Figura 16 – Ejemplo de comunicación en el sistema. ....	90
Figura 17 – Arquitectura Cliente-Servidor de 3 niveles .....	91
Figura 18 – Flujo de control del MVC en la aplicación. ....	92

Figura 19 – Diagrama de componentes. ....	93
Figura 20 – Diagrama E/R. ....	98
Figura 21 – Aplicación cliente. Bienvenida y selección de comensales .....	106
Figura 22 – Ejemplo de carta tradicional en papel de un restaurante.....	107
Figura 23 – Esquema de la interfaz de la carta. ....	108
Figura 24 – Aplicación cliente. Carta del restaurante. ....	109
Figura 25 – Transición al iterar entre Info y Opciones de un plato.....	110
Figura 26 – Aplicación cliente. Interfaz de eliminar ingredientes.....	111
Figura 27 – Aplicación cliente. Eliminar ingrediente del plato.....	112
Figura 28 – Selección del botón Ver Pedido desde la interfaz de la carta. ....	113
Figura 29 – Esquema de la interfaz de pedidos en la aplicación cliente.....	114
Figura 30 – Aplicación Cliente. Vista de Pedido en curso. ....	115
Figura 31 – Ventana emergente con el comentario introducido para el cocinero.....	116
Figura 32 – Aplicación Cliente. Mensaje de confirmación al realizar un pedido. ....	117
Figura 33 – Aplicación Cliente. Vista de Pedido realizado. ....	118
Figura 34 – Mensaje de confirmación a una valoración de plato recibida. ....	118
Figura 35 – Mensaje de confirmación para pedir la cuenta del pedido.....	119
Figura 36 – Mensaje de despedida y agradecimiento. ....	119
Figura 37 – Pantalla de Inicio. ....	120
Figura 38 – Aplicación Manager. Interfaz de Pedidos en Curso.....	121
Figura 39 – Aplicación Manager. Vista de pedidos en curso al pedirse la cuenta. ....	122
Figura 40 – Esquema de la interfaz Historial de búsqueda de la aplicación Manager.....	123
Figura 41 – Aplicación Manager. Vista de historial de pedidos. ....	124
Figura 42 – Mensaje de confirmación al imprimir la cuenta.....	124
Figura 43 – Aplicación Manager. Selección de fecha en filtro de búsqueda. ....	125
Figura 44 – Diagrama de navegación de la aplicación cliente.....	126
Figura 45 – Diagrama de navegación de la aplicación manager. ....	127
Figura 46 – Código del método <i>findAll</i> correspondiente a DishDAO .....	129
Figura 47 – Extracto de <i>findAll</i> . Código de conexión a base de datos. ....	130
Figura 48 – Extracto de <i>findAll</i> . Código de ejecución de query.....	130
Figura 49 – Extracto de código del método <i>create</i> correspondiente a <i>DishDao</i> .....	131
Figura 50 – Código de la clase <i>DishesResource</i> con métodos GET y POST.....	132
Figura 51 – Tratamiento de peticiones HTTP de tipo GET .....	133
Figura 52 – Tratamiento de peticiones HTTP de tipo POST .....	134
Figura 53 – Tratamiento de peticiones HTTP de tipo PUT .....	135
Figura 54 – Entidad <i>Mesa</i> y entidad <i>MesaList</i> . ....	136
Figura 55 – Método que prepara la llamada para obtener el listado de Cuentas. ....	137
Figura 56 – Método que prepara la llamada para insertar .....	138
Figura 57 – Código de recuperación de información en el Gestor REST.....	139
Figura 58 – Código de inserción de información en el Gestor REST. ....	140
Figura 59 – Diagrama de Gantt para la planificación del proyecto.....	153
Figura 60 – Jerarquía de directorios de los archivos de configuración .....	189

# Índice de tablas

Tabla 1 – Comparativa de características entre iPad mini, iPad 2 y iPad.....	37
Tabla 2 – Comparativa de características entre Samsung Galaxy tab 10.1, Motorola Xoom y Asus Eee Pad Transformer. ....	38
Tabla 3 – Comparativa entre el servidor web Tomcat 7 y el servidor de aplicaciones JBoss. ....	49
Tabla 4 – Comparativa entre los sistemas gestores de base de datos MySQL, SQL Server y Oracle Database. ....	50
Tabla 5 – Requisito de usuario UR_CC_01 .....	53
Tabla 6 – Requisito de usuario UR_CC_02 .....	54
Tabla 7 – Requisito de usuario UR_CC_03 .....	54
Tabla 8 – Requisito de usuario UR_CC_04 .....	54
Tabla 9 – Requisito de usuario UR_CC_05 .....	54
Tabla 10 – Requisito de usuario UR_CC_06 .....	55
Tabla 11 – Requisito de usuario UR_CC_07 .....	55
Tabla 12 – Requisito de usuario UR_CC_08 .....	55
Tabla 13 – Requisito de usuario UR_CC_09 .....	55
Tabla 14 – Requisito de usuario UR_CC_10 .....	56
Tabla 15 – Requisito de usuario UR_CC_11 .....	56
Tabla 16 – Requisito de usuario UR_MC_01 .....	56

Tabla 17 – Requisito de usuario UR_MC_02 .....	56
Tabla 18 – Requisito de usuario UR_MC_03 .....	57
Tabla 19 – Requisito de usuario UR_MC_04 .....	57
Tabla 20 – Requisito de usuario UR_MC_05 .....	57
Tabla 21 – Requisito de usuario UR_MC_06 .....	57
Tabla 22 – Requisito de usuario UR_MC_07 .....	58
Tabla 23 – Requisito de usuario UR_CR_01 .....	58
Tabla 24 – Requisito de usuario UR_CR_02 .....	58
Tabla 25 – Requisito de usuario UR_CR_03 .....	59
Tabla 26 – Requisito de usuario UR_CR_04 .....	59
Tabla 27 – Requisito de usuario UR_MR_01 .....	59
Tabla 28 – Requisito de usuario UR_MR_02 .....	59
Tabla 29 – Caso de uso UC_C_01 .....	61
Tabla 30 – Caso de uso UC_C_02 .....	62
Tabla 31 – Caso de uso UC_C_03 .....	62
Tabla 32 – Caso de uso UC_C_04 .....	63
Tabla 33 – Caso de uso UC_C_05 .....	63
Tabla 34 – Caso de uso UC_C_06 .....	64
Tabla 35 – Caso de uso UC_C_07 .....	65
Tabla 36 – Caso de uso UC_C_08 .....	65
Tabla 37 – Caso de uso UC_C_09 .....	66
Tabla 38 – Caso de uso UC_C_10 .....	67
Tabla 39 – Caso de uso UC_C_11 .....	67
Tabla 40 – Caso de uso UC_C_12 .....	68
Tabla 41 – Caso de uso UC_M_01 .....	69
Tabla 42 – Caso de uso UC_M_02 .....	69
Tabla 43 – Caso de uso UC_M_03 .....	70
Tabla 44 – Caso de uso UC_M_04 .....	70
Tabla 45 – Caso de uso UC_M_05 .....	71
Tabla 46 – Caso de uso UC_M_06 .....	72
Tabla 47 – Requisito de usuario SR_NFC_01 .....	74
Tabla 48 – Requisito de usuario SR_NFC_02 .....	74
Tabla 49 – Requisito de usuario SR_NFC_03 .....	74
Tabla 50 – Requisito de usuario SR_NFC_04 .....	75
Tabla 51 – Requisito de usuario SR_NFC_05 .....	75
Tabla 52 – Requisito de usuario SR_NFC_06 .....	75
Tabla 53 – Requisito de usuario SR_NFC_07 .....	76
Tabla 54 – Requisito de usuario SR_RFC_08 .....	76
Tabla 55 – Requisito de usuario SR_RFC_09 .....	76
Tabla 56 – Requisito de usuario SR_RFC_10 .....	76
Tabla 57 – Requisito de usuario SR_RFC_11 .....	77
Tabla 58 – Requisito de usuario SR_RFC_12 .....	77
Tabla 59 – Requisito de usuario SR_RFC_13 .....	77
Tabla 60 – Requisito de usuario SR_NFC_14 .....	78
Tabla 61 – Requisito de usuario SR_RFC_15 .....	78

Tabla 62 – Requisito de usuario SR_RFC_16 .....	79
Tabla 63 – Requisito de usuario SR_RFC_17 .....	79
Tabla 64 – Requisito de usuario SR_RFC_18 .....	79
Tabla 65 – Requisito de usuario SR_NFC_19 .....	80
Tabla 66 – Requisito de usuario SR_RFC_20 .....	80
Tabla 67 – Requisito de usuario SR_RFC_21 .....	81
Tabla 68 – Requisito de usuario SR_RFC_22 .....	81
Tabla 69 – Requisito de usuario SR_RFC_23 .....	82
Tabla 70 – Requisito de usuario SR_RFC_24 .....	82
Tabla 71 – Requisito de usuario SR_RFC_25 .....	82
Tabla 72 – Requisito de usuario SR_RFC_26 .....	82
Tabla 73 – Requisito de usuario SR_NFM_01 .....	83
Tabla 74 – Requisito de usuario SR_NFM_02 .....	83
Tabla 75 – Requisito de usuario SR_NFM_03 .....	83
Tabla 76 – Requisito de usuario SR_RFM_04 .....	84
Tabla 77 – Requisito de usuario SR_RFM_05 .....	84
Tabla 78 – Requisito de usuario SR_RFM_06 .....	84
Tabla 79 – Requisito de usuario SR_NFM_07 .....	84
Tabla 80 – Requisito de usuario SR_RFM_08 .....	85
Tabla 81 – Requisito de usuario SR_RFM_09 .....	85
Tabla 82 – Requisito de usuario SR_NFM_10 .....	85
Tabla 83 – Requisito de usuario SR_NFM_11 .....	86
Tabla 84 – Requisito de usuario SR_RFM_12 .....	86
Tabla 85 – Requisito de usuario SR_RFM_13 .....	86
Tabla 86 – Requisito de usuario SR_RFM_14 .....	87
Tabla 87 – Requisito de usuario SR_RFM_15 .....	87
Tabla 88 – Requisito de usuario SR_RFM_16 .....	87
Tabla 89 – Requisito de usuario SR_RFM_17 .....	88
Tabla 90 – Requisito de usuario SR_RFM_18 .....	88
Tabla 91 – Matriz de trazabilidad de requisitos del cliente de restaurante. ....	89
Tabla 92 – Matriz de trazabilidad de requisitos del manager .....	89
Tabla 93 – Componente Gestor de pedidos de cliente .....	94
Tabla 94 – Componente Visualizador de pedidos de cliente .....	94
Tabla 95 – Componente Gestor de comentarios .....	95
Tabla 96 – Componente Visualizador de comentarios de cliente .....	95
Tabla 97 – Componente Gestor de comunicación de cliente .....	95
Tabla 98 – Componente Gestor de comunicación manager .....	96
Tabla 99 – Componente Visualizador de pedidos de manager .....	96
Tabla 100 – Componente Filtro de búsqueda .....	96
Tabla 101 – Componente Gestor de pedidos activos .....	97
Tabla 102 – Matriz de trazabilidad componentes y UR. ....	97
Tabla 103 – Prueba de funcionalidad PF_01 .....	142
Tabla 104 – Prueba de funcionalidad PF_02 .....	142
Tabla 105 – Prueba de funcionalidad PF_03 .....	142
Tabla 106 – Prueba de funcionalidad PF_04 .....	143

Tabla 107 – Prueba de funcionalidad PF_05 .....	143
Tabla 108 – Prueba de funcionalidad PF_06 .....	143
Tabla 109 – Prueba de funcionalidad PF_07 .....	144
Tabla 110 – Prueba de funcionalidad PF_08 .....	144
Tabla 111 – Prueba de funcionalidad PF_09 .....	144
Tabla 112 – Prueba de funcionalidad PF_10 .....	144
Tabla 113 – Prueba de funcionalidad PF_11 .....	145
Tabla 114 – Prueba de funcionalidad PF_12 .....	145
Tabla 115 – Prueba de funcionalidad PF_13 .....	145
Tabla 116 – Prueba de funcionalidad PF_14 .....	146
Tabla 117 – Prueba de funcionalidad PF_15 .....	146
Tabla 118 – Prueba de funcionalidad PF_16 .....	146
Tabla 119 – Prueba de funcionalidad PF_17 .....	146
Tabla 120 – Prueba de funcionalidad PS_01 .....	147
Tabla 121 – Prueba de funcionalidad PS_02 .....	148
Tabla 122 – Prueba de funcionalidad PS_03 .....	148
Tabla 123 – Prueba de funcionalidad PS_04 .....	148
Tabla 124 – Matriz de trazabilidad de funcionalidad .....	150
Tabla 125 – Planificación de actividades .....	152
Tabla 126 – Desglose del coste de personal .....	155
Tabla 127 – Desglose del coste de Hardware .....	156
Tabla 128 – Desglose del coste de Software .....	156



# Capítulo 1

## Introducción y Objetivos

---

En este capítulo se tratan los objetivos y motivación que han llevado a la realización del proyecto, presentando una visión global del mismo e introduciendo el proceso de desarrollo desde su etapa más temprana. Para facilitar la tarea del lector, también se dedica un apartado para comentar brevemente el contenido de cada capítulo, a modo de resumen de cada uno de ellos.

## 1.1 Visión general

El objetivo del proyecto consiste en el desarrollo de un sistema que dé soporte al envío de pedidos en un restaurante por parte de los clientes, y a su vez a la recepción de los mismos desde la aplicación del manager que esté al cargo de la gestión de pedidos en el establecimiento.

La idea principal reside en almacenar la información acerca de un pedido realizado por un cliente, a la cual el manager del restaurante podrá acceder y consultar en tiempo real para poder organizar los pedidos en cocina. De igual manera, el manager podrá gestionar los pedidos bajo una serie de operaciones provistas por la aplicación, que también se guardarán en base de datos para futuras consultas.

El proyecto contará con dos módulos o aplicaciones, correspondientes a los roles contemplados en el proyecto, siendo estos *cliente* y *manager*. Cada aplicación contará con una interfaz exclusiva diseñada para facilitar el uso y manejo de la información por parte de cada rol, evitando mostrar información innecesaria que dificulte la navegación y uso de la misma.

Se cuenta también con un servidor dedicado única y exclusivamente a la recepción, tratamiento y envío de peticiones y respuestas tanto de cliente como de manager, que da soporte al acceso de la información almacenada.

## 1.2 Motivación

La tecnología está en constante evolución y a día de hoy son muchos los sectores que la han aprovechado para agilizar tareas en las que el ser humano puede ser prescindible, permitiendo una mayor eficiencia en cuestión de recursos y tiempo.

La hostelería lleva tiempo integrando ciertas tecnologías que permitan una mayor rapidez a la hora de tomar y gestionar pedidos. Han sido muchos los avances introducidos en este campo para conseguir dicho fin, pero sin duda queda un largo recorrido para agilizar este proceso y a la vez mejorar la experiencia del cliente.

Basándonos en ello, se dan dos motivaciones principales para la realización de este proyecto.

Una de ellas es el poder reducir el tiempo de espera de un cliente que llega a un restaurante y quiere realizar su pedido. El problema que suele darse es que los recursos, en este caso camareros, son limitados y el trabajo se acumula rápidamente: por un lado tienen que tomar nota y por otro repartir la comida; esto lleva a que los clientes tengan que esperar un tiempo indefinido a que un camarero pueda atenderles, traduciéndose en malestar generalizado tanto para el cliente que espera, como para el camarero que se ve desbordado de trabajo.

Ante este problema, el proyecto se idea para reducir la espera por parte del cliente y agilizar el proceso de realización de pedidos, liberando al camarero de la tarea de tomar nota, y pudiendo a su vez agilizar el reparto de pedidos entre las mesas.

La segunda motivación aparece al haber comprobado que con frecuencia existe cierta desorganización en la cocina cuando se tienen varios pedidos en espera: los pedidos se pierden, se toma nota incorrecta de las consumiciones, no se sabe si un plato ha sido enviado al comedor ya, se equivocan pedidos y mesas...

El proyecto pretende ser una ayuda para la persona que dirige la cocina y el comedor, proporcionando acceso inmediato a todos los pedidos recibidos desde el restaurante junto a los detalles relevantes del mismo. De esta manera, dirigir la cocina y tener claro todo lo que ocurre en el restaurante se convierte en una tarea más simple, al estar toda la información visible y actualizada en todo momento.

## 1.3 Objetivos

El objetivo del proyecto es la creación de un entorno que permita a un usuario hacer de forma autónoma un pedido en un restaurante a través de un dispositivo móvil y que este sea recibido y accedido en tiempo real por el responsable de gestión de pedidos del restaurante.

Este objetivo general engloba a su vez otros objetivos, los cuales son propuestos a continuación y permitirán el desarrollo del prototipo de la aplicación para la completa consecución del objetivo principal del proyecto:

- Desarrollo de la aplicación para el rol *cliente* que permita ver la carta del restaurante y realizar su pedido.
- Desarrollo de la aplicación para el rol *manager* que muestre la información de los pedidos realizados en el restaurante.
- Tratamiento y almacenamiento de toda la información intercambiada para poder ser accedida en cualquier momento.

Para poder conseguir estos objetivos, se plantea una serie de sub-objetivos más específicos que atienden a los requisitos de cada caso, como son:

- Ambas interfaces, pero con especial importancia la de cliente, deben ser intuitivas y fáciles de usar para mejorar la experiencia del usuario.
- Ha de darse al cliente la opción de modificar el pedido tal y como un camarero real permitiría hacer tomando nota de sus preferencias.
- El manager debe tener acceso inmediato a la información enviada por los clientes y tenerla continuamente a la vista para la organización de la cocina y el restaurante. A la vez, debe poder consultar y operar sobre pedidos activos y pedidos finalizados.
- Las dos aplicaciones deben mostrar tolerancia a errores cometidos por el usuario y debe estar preparada para responder correctamente a cualquier error sin interferir en el funcionamiento normal de la misma.

## 1.4 Fases de desarrollo

El desarrollo del proyecto se puede resumir en 6 fases.

La primera de estas fases coincide con el estudio de las tecnologías, plataformas y productos existentes, lo que permitirá llevar a cabo un estudio de viabilidad del sistema. En base a la información obtenida, en esta fase se decide la plataforma a utilizar en el desarrollo del proyecto y se analiza la arquitectura y características de la plataforma.

Una vez familiarizados con la plataforma, en la fase dos se realiza un análisis de requisitos que el sistema deberá cumplir. Para ello, se realizan varias entrevistas a clientes y se investiga sobre las necesidades y problemas en cocinas de restaurantes. De esta manera, es sencillo obtener los requisitos a través del análisis de esta información.

En la fase tres, teniendo los requisitos establecidos, se procede al diseño de la arquitectura del sistema estableciendo los componentes que cubrirían cada requisito del sistema. Lo que esto permite es asegurar que todas las necesidades recogidas en requisitos quedan recogidas en el proyecto.

En este punto se disponen de todos los datos necesarios para comenzar la programación, por lo que la fase cuatro se corresponde con la implementación de código.

Una vez realizada la parte de implementación, comienza la fase de realización de pruebas a través de un conjunto de tests, de modo que se depure el funcionamiento del sistema, perfilando comportamientos y solucionando errores.

Por último, la sexta fase consiste en la elaboración del actual documento.

## 1.5 Estructura del documento

Este apartado se destina a explicar brevemente los puntos desarrollados en los distintos capítulos del documento.

### **Capítulo 1.**

Pretende dar una visión global del proyecto, introduce los objetivos, las motivaciones y los acrónimos usados en el documento para una mejor comprensión por parte del lector.

### **Capítulo 2.**

Se analiza la evolución de la tecnología en diferentes ámbitos y, en base a ello y a las expectativas de futuro, se realiza la elección de plataforma de desarrollo y de las tecnologías para el desarrollo de la aplicación.

### **Capítulo 3.**

Este capítulo describen los procesos de análisis de requisitos, que definen las necesidades que el sistema debe solucionar, de decisiones de diseño tomadas para el desarrollo de interfaces y de implementación, donde se describen las características más destacables de código brevemente.

### **Capítulo 4.**

Detalla la planificación que se quiere seguir a lo largo de las distintas fases del proyecto, así como el presupuesto del sistema completo de acuerdo con la duración estimada.

### **Capítulo 5.**

En este capítulo se habla, de forma subjetiva, de las conclusiones llegadas una vez terminado el desarrollo del proyecto, así como de las líneas futuras de investigación y desarrollo que se seguirán para mejorar y completar la funcionalidad del sistema actual.

# Capítulo 2

## Estado de la cuestión

---

Este capítulo tiene como objetivo analizar las tendencias y avances tecnológicos para determinar el mejor enfoque posible para el proyecto. Solo haciendo un análisis del entorno actual, las tecnologías emergentes y los productos existentes en el mercado, será posible crear un software innovador y atractivo al usuario, capaz de cubrir las carencias de los productos actuales.

## 2.1 Evolución de la tecnología

### 2.1.1 Evolución de la tecnología en la hostelería

La hostelería es un sector muy exigente que requiere invertir muchas horas de trabajo y una dedicación casi plena, ya que sus horarios de apertura al público son extremadamente amplios.

Para permitir un mejor aprovechamiento de tiempo, recursos y esfuerzo, a lo largo del tiempo se han diseñado diferentes herramientas y máquinas que podemos ver a diario en cualquier servicio de hostelería. Herramientas como cajas registradoras, TPVs, PDAs, programas de gestión y un largo etcétera, han supuesto una mejora en términos de calidad de vida del servicio hostelero.

Entre estas herramientas, los TPVs son los que mayor mejora y facilidad de trabajo supusieron a la hora de tomar nota de las consumiciones de la clientela.

Los TPVs son cajas registradoras que disponen de una pantalla táctil para que el camarero introduzca los pedidos tras tomar nota en una mesa.

Suele haber un terminal de este tipo por restaurante y también permite la impresión de la factura, siempre y cuando se disponga de una impresora para tal fin conectada al dispositivo TPV.

La facilidad de uso de estos terminales difiere dependiendo del software instalado, pero se caracteriza por ser muy intuitivo. La mayoría de ellos comparten la característica de mostrar en pantalla una gran cantidad de botones entre los que se encuentran todos los platos y bebidas que se pueden pedir en la carta del restaurante, así como opciones y funciones que el camarero puede realizar [4].



Figura 1 – Caja registradora con TPV e impresora de tickets.



En la última década ha sido cada vez más notable el avance de la tecnología al incluir, junto a los TPVs, dispositivos PDA para facilitar al camarero tomar nota de cada mesa. Con ello, se evita el paso de tener que introducir de nuevo toda la información del pedido en el TPV tras tomar nota con *boli y papel* en las mesas.

Los dispositivos PDA se comunican con el TPV a través de conexión Wi-Fi o Bluetooth, enviando la información en tiempo real desde la misma mesa donde se realiza el pedido hasta cocina.



Figura 2 – Dispositivo PDA para realizar pedidos.

Debido a la pequeña dimensión de la pantalla de estos dispositivos, no se puede acceder a toda la información de un vistazo como en el caso del TPV, por lo que los artículos de la carta se distribuyen en familias, que pueden ser accedidos y elegidos de forma sencilla e intuitiva.

En adición a estas herramientas, se han desarrollado diferentes tipos de software que dan soporte a la gestión íntegra de establecimientos hosteleros, y que pueden ser accedidos a través de páginas web, o bien mediante la instalación sobre ordenadores personales.

A medida que avanza la tecnología, aparecen nuevas facilidades y herramientas que mejoran la calidad de servicio y la experiencia del cliente en el campo de la hostelería. Poco a poco, la manera de hacer pedidos cambia, se agiliza y es más eficiente, permitiendo una mejor gestión de recursos y tiempo.

Pero aún quedan muchos pasos por dar.

## 2.1.2 Evolución de la tecnología de las comunicaciones

Cuando se habla de tecnología, se habla de un concepto muy grande en el que se enmarca tanto a dispositivos físicos como a productos software. Se podría pensar que hay cierta ambigüedad en dicha terminología al englobar los dos conceptos, pero nada más lejos de la realidad, pues a fin de cuentas la evolución de hardware y software no tendrían sentido, ni sería posible, si ambas no fueran de la mano.

El campo de los dispositivos móviles ha sido uno de los más rápidos en evolucionar durante los últimos años. Se define como dispositivo móvil [5] a aparatos de pequeño tamaño que cuentan con características como:

- Capacidades especiales de procesamiento.
- Conexión permanente o intermitente a una red.
- Memoria limitada.
- Diseños específicos para una función principal y versatilidad para el desarrollo de otras funciones.

Ello engloba a un sinnúmero de dispositivos, entre los que encontramos teléfonos móviles, smartphones, sistemas de navegación por GPS, PDAs, reproductores MP3, videoconsolas, ordenadores portátiles y tablets.

A pesar del amplio abanico de aparatos que abarca el término dispositivo móvil, en el presente documento limitaremos el espectro para referirnos a smartphones, y tablets, ya que son, sobre todo estos últimos, los que realmente nos interesan para el desarrollo del proyecto.



Figura 3 – Dispositivos móviles. De izquierda a derecha: PDA, GPS, Smartphone, Tablet, MP3, videoconsola.

En los últimos años, el crecimiento de la telefonía móvil ha aumentado considerablemente, sobre todo con la llegada de dispositivos inteligentes, siendo así se prevé que en 2014 la navegación en Internet a través de dispositivos móviles sobrepasará a la realizada en ordenadores portátiles y de sobremesa[6] .

Al mismo tiempo, el número de usuarios de estas tecnologías también crece a diario. En 2011 se aproximaba en 4.000 millones de personas en todo el mundo haciendo uso de dispositivos móviles, en 2012 la cifra subió a alrededor de los 5.250 millones y se prevé que para 2013 se alcancen casi los 6.000 millones de usuarios [7]. Lo cual es un porcentaje muy alto de la totalidad de 7.000 millones de habitantes de población mundial existente en la actualidad, cuya cifra fue alcanzada en 2011 [8].

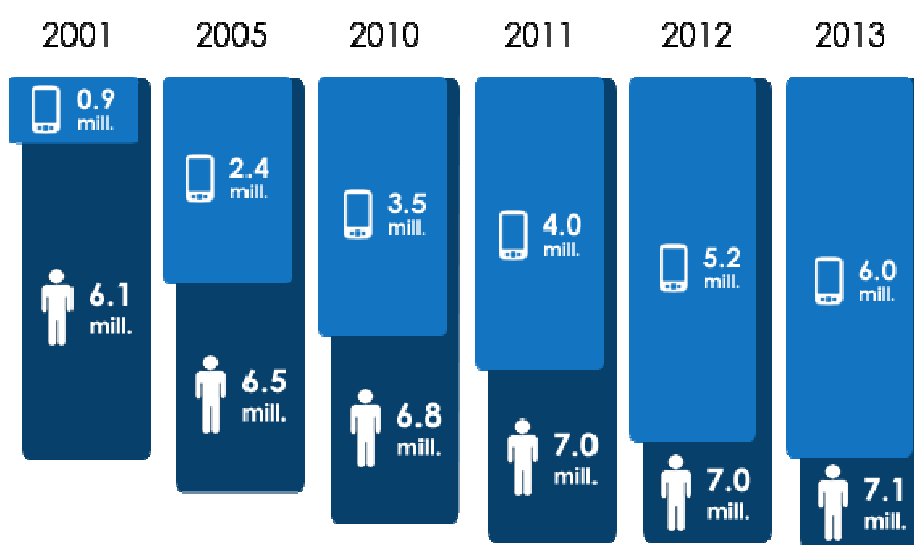


Figura 4 – Crecimiento del número de usuarios de dispositivos móviles.

El uso que se le da a los dispositivos móviles de hoy en día es muy diferente al que se le daba hace unos años. La introducción de teléfonos inteligentes y tablets ha llevado a que lo que menos se realice hoy en día, sean llamadas (que paradójicamente era lo único que permitían los orígenes de estos dispositivos), siendo la mensajería instantánea y el acceso a Redes Sociales los servicios que más han crecido [9] .

Es la capacidad de estos dispositivos la que permite el continuo desarrollo de la tecnología software, ofreciendo aplicaciones para todo tipo de situaciones y problemas de la vida diaria. Pero, para poder desarrollar una aplicación y que esta de buenos resultados, es necesario estudiar las distintas plataformas existentes en el mercado.

## 2.2 Análisis de la plataforma móvil

Hoy en día, existe a disposición del consumidor una amplia variedad de dispositivos móviles y marcas que los comercializan. Actualmente, las plataformas más extendidas entre los usuarios de dispositivos móviles y desarrolladores de aplicaciones son Android, iOS, Symbian, Blackberry y Windows Phone. Para conocerlas, a continuación se describe brevemente cada una de ellas:

- **Android** es el sistema operativo de Google. Está basado en Linux y su uso se enfoca a smartphones y tablets.

Dispone de una gran comunidad de desarrolladores de aplicaciones, las cuales pueden ser adquiridas a través del mercado Google Play directamente desde los dispositivos.



La programación de las aplicaciones se basa en Java. Desde su comienzo hasta el día de hoy, se han lanzado 4 versiones estables de este sistema operativo, siendo la más reciente la versión 4.1 (que fue lanzada en Octubre de 2012). No todas las versiones están disponibles para todos los dispositivos y compañías, y se depende del fabricante que da soporte al terminal para poder instalar versiones más recientes.

En Septiembre de 2012 el número de aplicaciones en Google Play superaba las 675.000 unidades y la cantidad de descargas ascendía hasta 25 mil millones [10] [11] .

El primer dispositivo con Android vio la luz en Octubre de 2008 y a finales de 2010 ya era líder del mercado de plataformas para Smartphone, desbancando a Symbian que había mantenido el liderazgo durante años. A nivel mundial, en el segundo cuatrimestre de 2012 alcanzó una cuota de mercado del 68%. A día de hoy hay 500 millones de dispositivos en el mundo activos que hacen uso de Android y se calcula que se hacen 1.3 millones de activaciones de dispositivos nuevos Android al día [12] .

- **iOS** es el sistema operativo desarrollado y distribuido por Apple. Fue lanzado para iPhone y iPod Touch en 2007 y ya podemos encontrarlo en otros dispositivos como iPad y Apple TV.



A diferencia de Windows Phone de Microsoft y Android de Google, Apple no permite la instalación de iOS en hardware que no pertenezca a Apple.

En el primer cuatrimestre de 2012, alcanzó una cuota de mercado del 23%. Dispone de tienda virtual, Apple Store, donde hay alrededor de 700.000 aplicaciones para iOS cuya suma de descargas asciende a los 30 mil millones. La versión más reciente de iOS es la 6.0.1 lanzada en Octubre de 2012 [13] .

- **Symbian** es un sistema operativo que en su inicio fue desarrollado por Symbian Ltd. para ser más tarde adquirida por Nokia.



Hasta 2010, Symbian era la plataforma más popular en el mundo de los Smartphones, pero el puesto le fue arrebatado por Android y poco a poco Symbian ha perdido adeptos entre sus filas, tanto de usuarios como de desarrolladores. Además de a la aparición de Android, este descenso de popularidad se debió a que en 2011 Nokia anunció que se asociaría con Microsoft, lo que llevó a la adaptación de Windows Phone en los dispositivos Nokia [14] .

- **Blackberry OS** es un sistema operativo desarrollado por Research In Motion (RIM) para dispositivos Blackberry.



Hasta 2010, este sistema operativo estaba únicamente orientado a smartphones, pero en septiembre de ese mismo año RIM anunció una nueva plataforma que sería integrada en su nuevo dispositivo tablet, el Blackberry PlayBook. Posee un market para descargar aplicaciones específicas para esta plataforma, llamado App World. En él existen alrededor de 99.500 aplicaciones, siendo el 25% de ellas específicas para la versión tablet [15] [16] .

- **Windows Phone** es el sistema operativo desarrollado por Microsoft. Es el sucesor de la plataforma Windows Mobile, pero a pesar de ello, son incompatibles y smartphones que han funcionado con Windows Mobile hasta ahora, no pueden ser actualizados a Windows Phone.



Del mismo modo, Windows Phone no ofrece compatibilidad con aplicaciones de Windows Mobile.

A diferencia de Windows Mobile, que estaba orientado a un mercado profesional, Windows Phone está enfocado a un mercado mucho más comercial, apto para el uso día a día de cualquier tipo de usuario. Windows Phone Store es el market virtual de esta plataforma y en ella se almacenan alrededor de 120.000 aplicaciones para Windows Phone. La versión más reciente de este sistema operativo es Windows Phone 8, que se dio a conocer a finales de Octubre de 2012 [17] [18] .

### 2.2.1 Criterios de selección de plataforma móvil

Conociendo los cinco sistemas operativos principales en el mercado, se debe buscar qué factores llevarán a tomar una decisión sobre la plataforma a utilizar. Para ello, se analizan las necesidades en el desarrollo del proyecto teniendo en cuenta que:

- **Punto 1:** No se trata de una aplicación a descargar de forma masiva por un gran número de usuarios, sino que está destinada a ser instalada en restaurantes, donde existirán varios dispositivos de las mismas características.
- **Punto 2:** El dispositivo ha de contar con una pantalla lo suficientemente grande como para poder ver detalladamente una carta de restaurante y los detalles que se deban añadir.
- **Punto 3:** El proyecto estará en constante evolución para mejorar la experiencia del usuario, por lo que una plataforma que crezca a su vez y permita el desarrollo de nuevas ideas e inclusión de nuevas tecnologías es completamente recomendable.

Estos tres puntos son esenciales y de ellos se pueden obtener varias conclusiones, que marcarán una guía para poder elegir la plataforma más recomendable de acuerdo con las características del proyecto. Estas conclusiones son:

- **Conclusiones del Punto 1:**

- A un usuario particular puede preocuparle que una aplicación esté o no diseñada para su terminal y que por esa razón no pueda acceder a ella, pero en este caso no es un problema ya que el restaurante contrata la aplicación independientemente del dispositivo (el dispositivo ha de ser adquirido de cualquier modo).
- El precio del dispositivo ha de ser competitivo, ya que habrá varias réplicas en el restaurante y la oferta de la aplicación ha de incluir aplicación + dispositivos. Si el precio de la oferta se ve disparado por incluir terminales caros (en relación necesidad/precio), las probabilidades de que no se venda son muy altas.

- **Conclusiones del Punto 2:**

- Hasta ahora se ha hablado de smartphones y tablets, pero por las características del proyecto, las dimensiones de pantalla de una tablet son idóneas para el desarrollo de la aplicación.

- **Conclusiones del Punto 3:**

- Se necesita una plataforma actual y en constante evolución. Ello solo se puede conseguir si el número de usuarios de la tecnología crece y existe una comunidad que de soporte tanto a los dispositivos como a las aplicaciones.
- Se ha de tener especial cuidado con sistemas operativos que no permitan compatibilidad con versiones anteriores ya que, con vistas al futuro, se podría dar el caso de que los dispositivos elegidos no pudieran ser utilizados si se actualiza a versiones más recientes del sistema operativo.

Del estudio de las prestaciones que necesitamos, a través del punto 2 concluimos que la plataforma ha de ser una tablet que permita al usuario ver de forma clara lo que está haciendo, lo que puede hacer y lo que ha hecho, de un solo vistazo. Por ello, Symbian es la primera plataforma a descartar, ya que a día de hoy no hay tablets que operen con este sistema operativo [19].

Blackberry OS por ahora solo puede utilizarse sobre el dispositivo tablet BlackBerry PlayBook. Debido a que el dispositivo tiene una pantalla de únicamente 7 pulgadas, desechamos la opción al ser demasiado pequeña para la aplicación que queremos desarrollar.



Figura 5 – Diferentes dispositivos Tablet. De izquierda a derecha y adelante a atrás: Apple iPad 2 (negro y blanco), BlackBerry PlayBook, Samsung Galaxy tab 10.1 y Samsung Galaxy tab 8.9

La segunda conclusión del punto 3 trata un tema importante, ya que como se vio anteriormente, Windows Phone no es compatible con la versión predecesora del sistema operativo, Windows Mobile. Con vistas al futuro se podría pensar que esto pudiera volver a ocurrir y la aplicación a desarrollar quedaría obsoleta, tanto en dispositivo como en tecnología si no se diera cobertura a partir de nuevas versiones. Es un riesgo que no se quiere correr, sobre todo cuando existen más plataformas para elegir al alcance.

Los dos candidatos finales son Android y iOS. Para poder continuar, ahora se ha de tener en cuenta el dispositivo sobre el que se ejecutará cada sistema operativo.



## 2.2.2 iOS

Puede ejecutarse sobre los dispositivos iPad, iPad mini y iPad 2. Las características más significativas de los 3 dispositivos se ven a continuación [21]:

	iPad mini	iPad 2	iPad
Dimensiones (cm)	20 / 13,47 / 0,72	24,12 / 18,57 / 0,88	24,12 / 18,57 / 0,94
Resolución pantalla	1024 x 768 a 163 pp.	1024 x 768 a 132 pp.	2.048 x 1.536 a 264 pp.
Tamaño diagonal pantalla (pulgadas)	7,9	9,7	9,7
Procesador	A5 dual core	A5 dual core	A6X dual-core con procesador gráfico de cuatro núcleos
Precio	<b>Wi-Fi</b> 16 GB - 329 € 32 GB - 429 € 64 GB - 529 €	<b>Wi-Fi</b> 16 GB - 479 € 32 GB - 579 €	<b>Wi-Fi</b> 16 GB - 499 € 32 GB - 599 € 64 GB - 699 €
	<b>Wi-Fi + 3G</b> 16 GB - 459 € 32 GB - 559 € 64 GB - 659 €	<b>Wi-Fi + 3G</b> 16 GB - 599 € 32 GB - 699 €	<b>Wi-Fi + 3G</b> 16 GB - 629 € 32 GB - 729 € 64 GB - 829 €

Tabla 1 – Comparativa de características entre iPad mini, iPad 2 y iPad

Por cuestiones de tamaño de pantalla, iPad mini se descarta. Las diferencias más significativas entre iPad 2 y iPad radican en el procesador, resolución de pantalla y precio.

Dado que los gráficos con los que trabaja la aplicación no son complejos ni requieren una calidad extraordinaria, queda descartado el iPad, ya que pagar un precio extra por algo que no se va a utilizar (ni requerir) inflaría el presupuesto final del proyecto.

Por tanto, el dispositivo **iPad 2 con 16GB de memoria interna y Wi-Fi + 3G** es el elegido para representar iOS en la comparativa final entre Android y iOS.

### 2.2.3 Android

Tiene una variedad de dispositivos mucho más amplia ya que muchas marcas han creado sus tablets confiando en esta plataforma. Encontramos dispositivos como Samsung Galaxy Tab 10.1, Motorola Xoom o Asus Eee Pad Transformer entre otros muchos [22] .

A principios de octubre de 2011, que fue cuando tuvo lugar la adquisición del dispositivo de prueba para el desarrollo del proyecto, en España solo se podían encontrar estos tres modelos disponibles. A pesar de que hoy en día se pueden encontrar modelos más actuales, se hará una comparativa entre estos tres dispositivos, pues tienen las características idóneas para el desarrollo del proyecto.

	Samsung Galaxy Tab 10.1 [23]	Motorola Xoom [24]	Asus Eee Pad Transformer [25]
Dimensiones (cm)	25,67 / 17,53 / 0,86	24,91 / 16,78 / 1,29	27,1 / 17,1 / 1,3
Resolución pantalla	1280 x 800 a 149 pp.	1280 x 800 a 150 pp.	1280 x 800 a 149 pp.
Tamaño diagonal pantalla (pulgadas)	10,1	10,1	10,1
Procesador	1GHz Dual core NVIDIA Tegra 2	1GHz Dual Core	1GHz Dual core NVIDIA Tegra 2
Precio	<b>Wi-Fi</b> 16 GB - 379 € 32 GB - 479 €  <b>Wi-Fi + 3G</b> 16 GB - 489 € 32 GB - 589 €	<b>Wi-Fi</b> 32 GB - 469 €  <b>Wi-Fi + 3G</b> 32 GB - 579 €	<b>Wi-Fi</b> 32 GB sin teclado - 499 € 32 GB con teclado - 599 €

Tabla 2 – Comparativa de características entre Samsung Galaxy tab 10.1, Motorola Xoom y Asus Eee Pad Transformer.

Como se aprecia, los tres modelos son prácticamente iguales en cuanto a prestaciones, así que la elección viene dada por confianza en la marca.

Por ello, el dispositivo elegido para representar a Android en la comparativa entre sistemas operativos y dispositivos es el **Samsung Galaxy Tab 10.1 de 16GB con Wi-Fi y 3G**.

## 2.2.4 Plataforma móvil seleccionada

Teniendo en cuenta las conclusiones del punto 1, 2 y 3, buscamos una plataforma en constante evolución, con una comunidad de desarrolladores y usuarios grande y activa. En el siguiente gráfico vemos como ha evolucionado el mercado de aplicaciones de Android y iOS durante los cuatro últimos años [26][27].

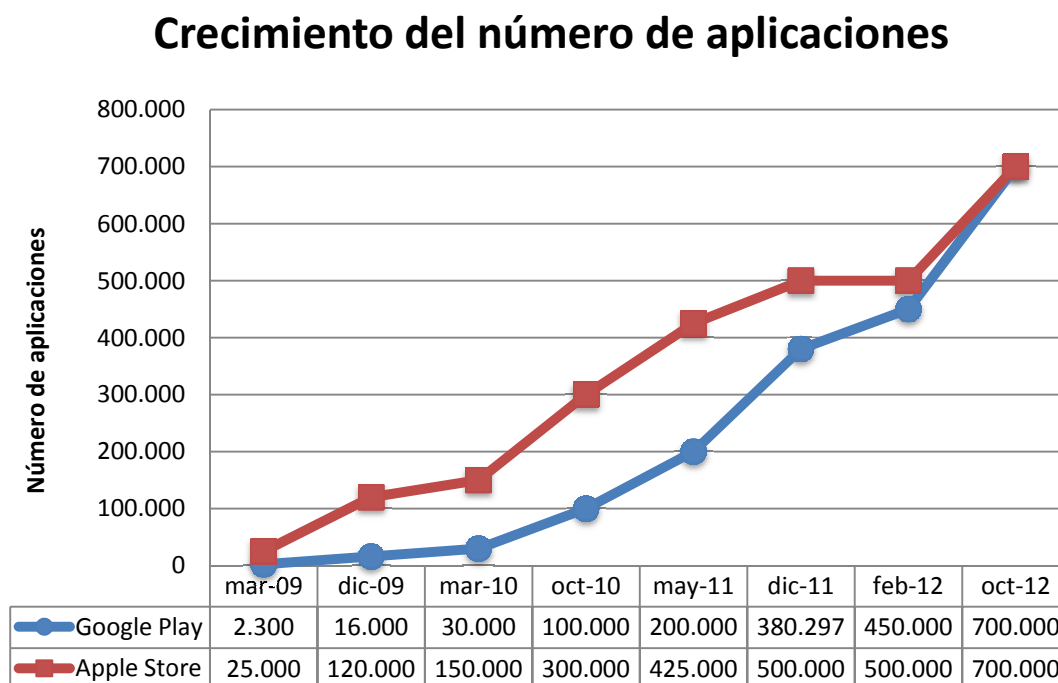


Figura 6 – Crecimiento del número de aplicaciones para Android y iOS desde 2009 a 2012.

Es apreciable que la cantidad de aplicaciones desarrolladas para Android crece de forma casi exponencial, siendo así que a pesar de que el market de iOS le lleva un año de ventaja, en octubre de 2012 alcanzaron el mismo volumen de aplicaciones. Anteriormente vimos que el número de descargas realizadas para las plataformas Android y iOS ascienden a 25 y 30 mil millones respectivamente

Si a eso le añadimos la cuota de mercado de la que hablamos con anterioridad (en el segundo cuatrimestre de 2012 la cuota de mercado de Android era del 68%, frente a la de iOS del 23%), todo apunta a que el crecimiento de Android será indiscutiblemente superior al de iOS en poco tiempo.

Esta razón lleva a elegir la plataforma **Android 3.2** para el desarrollo del proyecto, y el terminal **Samsung Galaxy Tab 10.1 de 16GB con Wi-Fi y 3G**, el dispositivo para la realización de pruebas.

## 2.3 Análisis de aplicaciones

Desde la llegada de Android a los smartphones, el número de aplicaciones en el market de Android ha crecido rápidamente. Se puede encontrar una aplicación para casi todo lo imaginable, y por supuesto, también para hacer pedidos desde el dispositivo móvil.

Aunque la cantidad de aplicaciones de este tipo es mucho mayor en Estados Unidos, la mayor parte de ellas están enfocadas desde el mismo punto de vista y las diferencias entre aplicaciones son mínimas.

En España existen menos aplicaciones de este tipo, pero son mucho más innovadoras y diferentes entre sí, además de que cuidan detalles que mejoran la experiencia del usuario gratamente.

A continuación se muestra una selección de las más interesantes con los datos más característicos de cada una.

### OrderingApps

Permite realizar pedidos a ciertos restaurantes y decidir a qué hora se quiere pasar a recoger. Se puede instalar sobre ciertos dispositivos de las plataformas Android y iOS [28] , siendo más frecuente para smartphone que para tablet. Solo está disponible en Estados Unidos y únicamente se pueden hacer pedidos a restaurantes que hayan publicado su carta en la aplicación.

Esta aplicación permite:

- Enviar o reenviar un pedido.
- Indicar a qué hora se quiere recoger el pedido.
- Encontrar la localización del restaurante.
- Enviar comentarios a redes sociales
- Ver si existen ofertas o descuentos en los restaurantes.

Tras realizar un pedido, al restaurante le llega la notificación por email, fax o a través de un portal de internet.

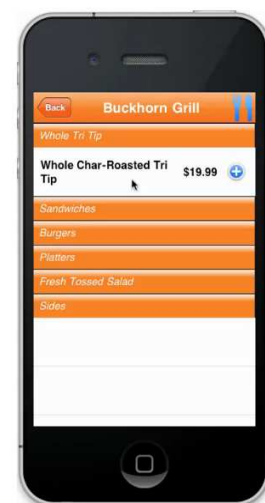


Figura 7 – Interfaz de OrderingApps

## Snapfinger

Esta aplicación es muy similar a OrderingApps en cuanto a funcionalidad ofrecida, pero el número de restaurantes sobre los que se puede realizar un pedido con Snapfinger es mucho mayor (entre los que se incluyen cadenas de restaurantes como Tony Roma's o Subway). La aplicación permite:

- Ver los restaurantes próximos al usuario para poder realizar el pedido a uno de ellos.
- Buscar por nombre del restaurante, tipo de cocina o plato que se quiere consumir.
- Reenviar un pedido hecho con anterioridad.
- Indicar a qué hora se recogerá el pedido en el restaurante.
- Pagar mediante tarjeta de crédito (aunque puede indicarse que se pagará en el restaurante al recogerlo).



Figura 8 – Interfaz de Snapfinger

Sólo está disponible en Norte América y se puede pedir a más de 500.000 restaurantes. Se puede utilizar bajo dispositivos Android y iOS y está especialmente diseñado para su uso en smartphones.

## Food2U

Food2U hasta ahora se trataba de un servicio web mediante el cual se podía pedir comida a domicilio desde su página en internet. Con la llegada de las nuevas tecnologías, se han sumado al movimiento Android y iOS, desarrollando su aplicación también para ser descargadas en dichas plataformas.

Comparte ciertas funcionalidades con las aplicaciones vistas anteriormente, como ver y elegir a qué restaurantes cercanos se puede hacer un pedido o realizarlo desde el dispositivo móvil desde cualquier lugar, pero Food2U se centra en el reparto de comida a domicilio, sin que el usuario tenga que ir a buscar el pedido al restaurante.

La aplicación muestra información como:

- Si un restaurante está abierto o cerrado en el momento de hacer el pedido.
- El tiempo de entrega de un pedido.
- El precio de entrega de los pedidos a domicilio.
- La calificación media hecha por los usuarios sobre el restaurante.
- El precio mínimo que tiene que alcanzar el pedido para que pueda enviarse a domicilio.



Figura 9 – Interfaz de Food2U

Tanto si el restaurante acepta como si rechaza el pedido realizado, se envía un email de notificación a los 5 minutos de haber realizado el pedido.

Esta aplicación sí está disponible en España, aunque solo en ciudades como Madrid o Barcelona será posible encontrar una cantidad aceptable de restaurantes para poder hacer un pedido [33].

## QlikBar

Se trata de una aplicación con diversas funcionalidades, todas ellas relacionadas con la hostelería pero distintas entre sí, entre las que cabe destacar:

- Sirve como guía para conocer la ubicación, la tipología, los horarios o el menú de los bares, restaurantes y cafeterías adheridos a la aplicación.
- Permite a los usuarios agregar a su red de amigos y consultar su actividad.
- Existe un sistema de puntos y recompensas al visitar locales de hostelería.
- En los locales adheridos, permite pedir desde el dispositivo.



Figura 10 – Interfaz de QlikBar

Esta aplicación es distinta a las anteriores, ya que está más enfocada a dar a conocer los restaurantes adheridos entre las redes sociales y localizar en qué bares o restaurantes están los contactos del usuario del dispositivo, que a hacer pedidos en restaurantes.

Está desarrollada, y por tanto disponible, en España aunque la cantidad de locales adherida a esta aplicación y la actividad en los mismos aun es escasa.

Está disponible para Android y iOS y tienen dos versiones: una gratis (con funcionalidad limitada) y otra Premium.

### **2.3.1 Comparativa**

Las aplicaciones vistas anteriormente, a pesar de tener finalidades similares, son muy distintas a la aplicación desarrollada aquí como proyecto de fin de carrera. Mientras dichas aplicaciones ofrecen lo visto con anterioridad, la que actualmente se desarrolla ofrece:

- Un diseño orientado a tablet en lugar de smpartphone, que permite una mejor experiencia al usuario al poder ver en pantalla mucha más información que en un Smartphone. Se evita tener que cambiar de pantalla continuamente, viendo la carta de restaurante como lo haría si estuviera impresa en papel.
- El cliente no ha de poseer un Smartphone, ni una tablet, ni una cuenta de usuario en la aplicación. El restaurante ofrece el dispositivo tablet al usuario, ya configurada para ser usada por el cliente.
- La aplicación está diseñada para hacer un pedido y tomarlo en el mismo restaurante. No está diseñada para recibir pedidos a domicilio ni recogerlos, sino para el consumo en el momento.
- Todos los comentarios realizados se ven en el mismo menú, sin necesidad de acceso a redes sociales, ayudando al cliente a tener una idea de la calidad del plato en el mismo momento.

Por estas razones, hasta el momento la competencia existente para nuestro proyecto en el ámbito tablet y en nuestro país es escasa, por lo que las expectativas de futuro de la aplicación a día de hoy son grandes.

## 2.4 Tecnología utilizada

En el capítulo 2.2 se hizo un análisis de los dispositivos móviles actuales y se obtuvieron ciertas conclusiones con las que finalmente se eligió Android como plataforma de desarrollo y la tablet Samsung Galaxy Tab 10.1 como dispositivo móvil sobre el que se ejecutará el proyecto.

Con ello quedaron establecidas las bases para poder comenzar las fases de análisis y desarrollo de la aplicación, pero no serán las únicas herramientas que ha utilizar.

En este apartado se enunciará y hará un breve resumen de las tecnologías, lenguajes de programación y frameworks empleados a lo largo de la fase de desarrollo, los cuales tienen que ver directamente con la programación del código. En siguientes capítulos veremos en detalle estas tecnologías y de qué manera nos han ayudado en el proyecto.

### 2.4.1 Herramientas básicas

Siempre hay una serie de herramientas y utilidades básicas necesarias en el desarrollo de cualquier proyecto. Este apartado recoge el conjunto de estas herramientas, con un breve detalle de sus características más relevantes.

#### ✓ Entorno de desarrollo: Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto que nos permitirá desarrollar sobre él toda la parte de programación de código.

Dado que plugin ADT que proporciona Android está especialmente desarrollado para Eclipse y proporciona una serie de herramientas que agilizan y facilitan el desarrollo del código, Eclipse se convierte en el entorno de desarrollo más recomendable para trabajar en este proyecto.

Trabajaremos con la versión *Eclipse Indigo IDE for Java EE Developers* [37] , que más adelante nos permitirá crear, además de los dos clientes, el servidor.



## ✓ **Lenguaje de desarrollo Java**

Java es un lenguaje de alto nivel orientado a objetos que se caracteriza por ser distribuido, interpretado y compilado al mismo tiempo, robusto, independiente de la arquitectura y portable.

Lo más frecuente es encontrar aplicaciones Android programadas en Java, y dado que se dispone de un amplio conocimiento del lenguaje y todo apunta a que es el lenguaje idóneo para realizar el proyecto, será el utilizado en el desarrollo.

Se empleará Java SE 7 y la versión 1.7.0 del JDK.

## ✓ **Android SDK**

El SDK de Android proporciona las librerías y herramientas para la programación de código Android, por lo que su instalación es imprescindible.

Se puede descargar el manager de instalación desde la página de desarrolladores Android [38], y él mismo se encarga de descargar todo lo necesario para instalar el SDK de la versión que se requiera, además de otros componentes necesarios.

## ✓ **Android ADT Plugin**

Es el conjunto de herramientas de desarrollo proporcionado por Android para la creación de aplicaciones en Eclipse [39].

Se trata de un IDE completo que extiende las capacidades de Eclipse, permitiendo crear la interfaz Android tanto con la introducción de elementos mediante drag-and-drop (arrastrar y soltar en la interfaz) como programáticamente (escribiendo el código), debuggear y testear la aplicación, exportar el paquete para su distribución, así como emular el proyecto sin necesidad de disponer de un dispositivo físico.

Es un plugin muy potente y realmente útil para realizar proyectos Android.

## 2.4.2 Herramientas específicas

Para el desarrollo del proyecto será necesario el uso de librerías, frameworks y herramientas que nos permitan aportar funcionalidad adicional y mucha más potencia de la permitida por las herramientas básicas.

Cada herramienta descrita a continuación ha llevado, detrás de su implantación en código, muchas horas de análisis y comparativas con respecto a herramientas similares. Para justificar las decisiones de elección, se hará una breve comparativa entre tecnologías.

### ✓ Servicio Web

Un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares para permitir el intercambio de datos entre aplicaciones, independientemente del lenguaje de programación o de la plataforma sobre la que se ejecuten.

A día de hoy, existen dos grandes protocolos que permiten la comunicación entre aplicaciones: SOAP y REST.

- **SOAP**

Previo al nacimiento de SOAP, se utilizaba RPC para la comunicación entre objetos. Este protocolo tenía como principal problema que no estaba diseñado para ser utilizado sobre HTTP con lo que, por norma, todo el tráfico era bloqueado por firewalls y proxys.

SOAP fue creado para que la comunicación entre aplicaciones fuera posible sobre HTTP, ya que cualquier servidor y navegador lo soporta y permite que la comunicación se produzca entre plataformas y lenguajes distintos.

El formato estándar de mensajes utilizado por SOAP es XML, que a pesar de poder leerse y permitir la detección de errores fácilmente, supone un procesamiento más lento que formatos de mensaje binarios.

- **REST**

Se ideó para evitar utilizar mecanismos complejos como RPC o SOAP al realizar comunicaciones entre aplicaciones.

La idea de REST es permitir el acceso a los recursos a través de identificadores. Cada recurso tiene su propio identificador, el cual es accedido a través de su propia URL, y para trabajar con dichos recursos se utilizan las operaciones estándar de HTTP (GET, PUT, POST, DELETE...).

Los objetos REST suelen representarse en XML y JSON.

Teniendo una visión general de ambos protocolos, a continuación se discuten sus características más relevantes de cara al proyecto.

- **Seguridad:** Es uno de los factores más interesantes a tener en cuenta.
  - REST dispone de *endpoints* predecibles y el contenido de una petición es un mensaje simple de HTTP. Aún así, un usuario que quisiera vulnerar la seguridad de un mensaje REST, debería entender la estructura de la URL utilizada en los intercambios.
  - SOAP necesita consumir un WSDL con el fin de conocer los *endpoints*. Por otro lado, SOAP dota a los objetos de información del formato entero del mensaje, por lo que se pueden realizar ataques específicos.

Ambos son igual de inseguros por sí mismos, por lo que la seguridad ha de reforzarse mediante la encriptación de comunicaciones, la autenticación y autorización de usuarios antes de procesar información y unos buenos hábitos de codificación para evitar ataques.

- **Complejidad:** REST nació con el propósito de eliminar la complejidad ofrecida por SOAP. Y lo hizo. La facilidad de uso de REST es indiscutible frente al hecho de necesitar crear mensajes complejos con SOAP.

Todo ello lleva a la conclusión de que **REST** es el protocolo que mejor se adapta a las necesidades del proyecto, haciendo muy sencillo tanto su uso como su integración con el resto de herramientas.

## ✓ API de creación de servicios web

Existen numerosas APIs que facilitan la creación de servicios web. Entre ellas destaca JAX-RS, que se trata de una API de Java para servicios web basados en protocolo REST, lo cual coincide con las líneas exactas del proyecto.

Se introdujo en la versión Java SE 5 para la simplificación del desarrollo y despliegue de servicios web, y se convirtió en parte oficial de Java a partir de la versión EE 6, lo que posibilita su uso directo sin necesidad de configuraciones adicionales. Se caracteriza por el uso de anotaciones para facilitar el mapeo de clases Java (POJO) a recursos web.

Para hacer uso de JAX-RS en el proyecto, necesitamos una implementación sobre este estándar. Entre los más comunes se encuentra Jersey, RESTEasy, CXF o Restlet. Todas ofrecen características similares, pero **Jersey**, a diferencia del resto, tiene una comunidad de usuarios desarrolladores muy amplia, lo que le confiere una madurez mayor a la del resto de implementaciones.

## ✓ Librería de conversión JSON

En la comunicación entre clientes y servidor, la información intercambiada mediante operaciones HTTP debe ser convenientemente mapeada, de manera que tanto al salir de un extremo, como al llegar a otro, ambos sepan de qué manera leer la información para utilizarla.

No es aconsejable el envío de objetos puros de Java al trabajar con REST, ya que es mucho más sencillo utilizar XML o JSON para realizar cualquier intercambio de información. Entre ambas opciones, JSON es la más sencilla tanto para implementar, como para enviar datos, ya que prescinde de las etiquetas propias de XML y utiliza exclusivamente una lista en la que toda la información se almacena ordenadamente.

Existen librerías específicas para convertir objetos Java a JSON y viceversa. Se utilizará la librería **GSON** por su sencillez de uso, ya que con las llamadas *toJson()* y *fromJson()* se consigue convertir de un objeto a otro, permitiendo además representaciones personalizadas de objetos.

## ✓ Servidor

Existen distintos tipos de servidores, donde cada cual desempeña funciones más o menos complejas y están orientados a unos u otros objetivos. A continuación se analizan los dos tipos de servidores más convenientes al proyecto: servidores web y servidores de aplicaciones.

Un servidor web tiene como función principal el envío y recuperación de peticiones mediante el protocolo HTTP, ofreciendo la capacidad de mapeo de URLs que, como hemos visto anteriormente, será imprescindible para la recuperación de información mediante el estándar JAX-RS.

Un servidor de aplicaciones proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas en él. Por lo general, la funcionalidad que ofrece es mayor que la de los servidores web, pero también lo es su complejidad de uso, administración, configuración y recursos utilizados.

Tomcat es el servidor web por excelencia. A partir de su versión 7 incluye funcionalidad propia de los servidores de aplicaciones, haciendo el distanciamiento entre ambos tipos de servidores cada vez menor. En cuanto a servidores de aplicaciones hay bastante más diversidad y se pueden encontrar a competidores como Glassfish o Gerónimo, pero por madurez del producto y mayor extensión de comunidad de usuarios, JBoss es el más popular entre los servidores de aplicaciones.

	Tomcat 7	JBoss
Compañía	Apache Software Foundation	Red Hat
Recursos utilizados	Bajo	Alto
Complejidad de administración	Baja	Alta
Funcionalidad ofrecida	Media	Alta
Compatibilidad con Java	Si	Si
Multiplataforma	Si	Si

Tabla 3 – Comparativa entre el servidor web Tomcat 7 y el servidor de aplicaciones JBoss.

Por las facilidades que ofrece y por cubrir perfectamente los requisitos del proyecto, se elige **Tomcat** como servidor web para el desarrollo del proyecto.

## ✓ Sistema gestor de base de datos.

Los sistemas gestores de bases de datos, también conocidos como SGBD, son tipos de software encargados de manejar los datos que se utilizarán en una aplicación de manera sencilla y estructurada.

Como objetivos principales de un SGBD se destaca la fiabilidad y seguridad, la abstracción de la información, la facilidad de manejo en las transacciones y la consistencia en datos, lo que facilita en gran medida la tarea del usuario al poder desentenderse de toda la parte de almacenamiento y recuperación de información.

La existencia de un gran número de sistemas gestores de bases de datos lleva a que hagamos el siguiente resumen de las características principales de los más usados hoy en día: MySQL, SQL Server y Oracle Database. Ello nos permitirá hacer una elección del más conveniente para el proyecto.

	MySQL	SQL Server	Oracle Database
Compañía	Desarrollado por Sun Microsystems. Adquirida por Oracle.	Microsoft	Oracle
Rendimiento	Alto	Alto	Muy Alto
Capacidad	Alto	Alto	Muy Alto
Concurrencia	Si	Si	Si
Licencia Libre	Si	No	No
Multiplataforma	Si	No	Si
Seguridad	Alto	Alto	Alto

Tabla 4 – Comparativa entre los sistemas gestores de base de datos MySQL, SQL Server y Oracle Database.

Es de apreciar que los tres sistemas tienen muy buenas características, aunque para el desarrollo del proyecto la que más nos conviene por ser multiplataforma, de licencia libre y con un rendimiento y capacidad ajustada a las necesidades, es **MySQL**.

# Capítulo 3

## Análisis, Diseño e Implementación

---

En este capítulo se especifican las fases de análisis de requisitos, diseño de la aplicación e implementación a nivel de código. En cada apartado, se tratan de los módulos de la aplicación cliente de restaurante, la aplicación manager y el servidor de forma separada. Al final del capítulo, se detallan algunas de las pruebas utilizadas para el testeo del sistema.

## 3.1 Análisis

### 3.1.1 Requisitos de Usuario

Este apartado recoge los requisitos de usuario, que se dividirán en dos categorías: requisitos de usuario de capacidad y requisitos de usuario de restricción

Para facilitar la lectura, los requisitos serán recogidos en tablas cuyo formato incluirá los siguientes campos:

- **Nombre:** Nombre identificativo del requisito. Ha de ser unívoco y se utilizará a modo de resumen de dicho requisito. No tiene que incluir detalles, pero sí ser descriptivo.
- **ID:** identificador unívoco del requisito de usuario. Su nomenclatura ha de seguir el formato: UR\_XY\_ZZ, donde:
  - X será sustituido por 'M' en caso de tratarse de los requisitos de usuario de la aplicación del manager, o 'C' si se trata de los requisitos de la aplicación que maneja el cliente del restaurante desde la mesa.
  - Y será sustituido por 'C' en caso de tratarse de un requisito de usuario de capacidad, o 'R' si se tratara de un requisito de restricción.
  - ZZ será sustituido por el número de requisito dentro de su categoría. Comenzará en 01 e incrementará a cada nuevo requisito en una unidad.
- **Descripción:** Descripción detallada del requisito. Todos los detalles han de incluirse aquí para no dar lugar a ambigüedades futuras.
- **Prioridad:** Indica la prioridad de implementación del requisito. Facilita la organización y carga de trabajo a llevar durante el desarrollo del proyecto. Puede tomar los valores *Baja*, *Media* y *Alta*.



- **Necesidad:** Indica como de necesaria es la introducción del requisito en el proyecto. Puede tomar los valores *Esencial* y *Deseable*.
- **Estabilidad:** Indica si el requisito puede ser objeto de modificaciones durante el ciclo de vida del proyecto, o si por el contrario se trata de un requisito que no ha de sufrir variaciones. El campo puede tomar los valores *Estable* e *Inestable*, dependiendo de si se da el segundo o primer caso, respectivamente.
- **Verificabilidad:** Establece con qué facilidad se puede comprobar que el requisito haya sido introducido en el proyecto. Los valores que puede tomar son *Alta*, *Media* y *Baja*.

### 3.1.1.1 Requisitos de Capacidad

Los requisitos de capacidad, indican qué funciones puede o tiene que hacer la aplicación para cumplir los propósitos requeridos para el sistema.

A continuación, se detallan los requisitos de capacidad para las aplicaciones cliente de restaurante y manager.

#### ❖ Cliente del Restaurante

BIENVENIDA Y COMENSALES			UR_CC_01
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario debe recibir un mensaje de bienvenida y poder especificar el número de comensales nada más entrar en la aplicación.		

Tabla 5 – Requisito de usuario UR\_CC\_01

VISTA DE PLATOS EN CATEGORÍA			UR_CC_02
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>El usuario debe poder elegir una categoría y ver el listado de platos existente en ella. Una vez seleccionado un plato, el cliente deberá tener acceso a la siguiente información:</p> <ul style="list-style-type: none"> <li>• Nombre del plato.</li> <li>• Imagen del plato.</li> <li>• Descripción del plato.</li> <li>• Precio del plato.</li> <li>• Valoración y comentarios del plato.</li> <li>• Apto para celíacos.</li> <li>• Apto para vegetarianos.</li> </ul>		

Tabla 6 – Requisito de usuario UR\_CC\_02

ELIMINAR INGREDIENTE			UR_CC_03
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>El usuario debe tener la opción de eliminar ingredientes de un plato, antes de añadirlo al pedido, siempre y cuando existan ingredientes que eliminar en dicho plato.</p>		

Tabla 7 – Requisito de usuario UR\_CC\_03

INTRODUCIR COMENTARIO PARA EL COCINERO			UR_CC_04
Prioridad	Alta	Necesidad	Deseable
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>El usuario debe tener la opción de introducir un comentario sobre un plato, con órdenes para el cocinero, antes de añadirlo al pedido.</p>		

Tabla 8 – Requisito de usuario UR\_CC\_04

AÑADIR PLATO A PEDIDO			UR_CC_05
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>El usuario debe tener la opción de añadir un plato a su pedido, tantas veces como desee.</p>		

Tabla 9 – Requisito de usuario UR\_CC\_05

ELIMINAR PLATO			UR_CC_06
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario puede reducir la cantidad de veces que se pide un plato o eliminar dicho plato del pedido, siempre y cuando el pedido no haya sido realizado.		

Tabla 10 – Requisito de usuario UR\_CC\_06

REALIZAR PEDIDO			UR_CC_07
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario debe tener la opción de realizar un pedido para empezar a prepararse en cocina, una vez haya añadido al menos un plato a dicho pedido. Después de enviar, debe tener la opción de seguir añadiendo platos y enviando pedidos a la cocina, a pesar de haberse enviado un pedido con anterioridad durante la misma sesión.		

Tabla 11 – Requisito de usuario UR\_CC\_07

COMENTAR Y VALORAR PLATO PEDIDO			UR_CC_08
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario debe tener la opción de comentar y valorar un plato, siempre y cuando haya sido añadido en su pedido actual, y el pedido haya sido enviado a la cocina para su preparación.		

Tabla 12 – Requisito de usuario UR\_CC\_08

VER VALORACIONES Y COMENTARIOS			UR_CC_09
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario ha de poder ver todas las valoraciones y comentarios realizados sobre los platos de la carta.		

Tabla 13 – Requisito de usuario UR\_CC\_09

VISUALIZAR PEDIDO			UR_CC_10
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario ha de poder visualizar tanto el pedido que aun no ha sido enviado a la cocina, como los ya enviados a la cocina durante la sesión actual.		

Tabla 14 – Requisito de usuario UR\_CC\_10

PEDIR CUENTA			UR_CC_11
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario debe poder pedir la cuenta y con ello terminar la sesión.		

Tabla 15 – Requisito de usuario UR\_CC\_11

## ❖ Manager

VER DETALLE DE PEDIDOS ACTIVOS			UR_MC_01
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>El usuario debe poder acceder a los pedidos activos en el restaurante, donde debe tener una visión completa de:</p> <ul style="list-style-type: none"> <li>• Listado de mesas que están esperando un pedido.</li> <li>• Número de mesa seleccionado.</li> <li>• Platos pedidos en la mesa seleccionada.</li> <li>• Cantidad de cada plato pedido en la mesa seleccionada.</li> <li>• Comentarios para cocina junto al plato (en caso de haber).</li> <li>• Precio total de la mesa seleccionada.</li> </ul>		

Tabla 16 – Requisito de usuario UR\_MC\_01

MARCAR PLATOS COCINADOS			UR_MC_02
Prioridad	Media	Necesidad	Deseable
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario debe poder marcar, sobre un pedido activo, qué platos van siendo enviados a los clientes.		

Tabla 17 – Requisito de usuario UR\_MC\_02

NOTIFICACIÓN DE CUENTA			UR_MC_03
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	El usuario debe ser notificado cuando una mesa pide la cuenta.		

Tabla 18 – Requisito de usuario UR\_MC\_03

MARCAR PAGADO Y TERMINAR PEDIDO			UR_MC_04
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	Una vez el cliente ha recibido y pagado la cuenta, el usuario ha de marcar el pedido como pagado. Si además todos los platos han sido preparados, el usuario debe tener la opción de terminar y archivar el pedido.		

Tabla 19 – Requisito de usuario UR\_MC\_04

ACTUALIZAR INTERFAZ			UR_MC_05
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	El usuario debe tener la opción de actualizar la interfaz para ver los pedidos nuevos que se realizan en el restaurante.		

Tabla 20 – Requisito de usuario UR\_MC\_05

ACCESO A HISTORIAL DE PEDIDOS			UR_MC_06
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>El usuario debe poder acceder a un historial de pedidos, donde se muestre:</p> <ul style="list-style-type: none"> <li>• Filtros de búsqueda.</li> <li>• Listado con pedidos terminados.</li> <li>• ID del pedido seleccionado.</li> <li>• Número de mesa desde donde se hizo el pedido.</li> <li>• Platos y cantidad de ellos pedidos.</li> <li>• Comentarios de los platos (si hubiera).</li> <li>• Fecha del pedido.</li> <li>• Precio final del pedido.</li> <li>• Icono si el pedido ha sido pagado.</li> </ul>		

Tabla 21 – Requisito de usuario UR\_MC\_06

BÚSQUEDA DE PEDIDOS			UR_MC_07
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	<p>El usuario debe poder buscar uno o varios pedidos introduciendo cero, uno o más filtros de búsqueda, que filtrarán por:</p> <ul style="list-style-type: none"> <li>• Número de mesa en la que se realizó el pedido.</li> <li>• Precio máximo y/o mínimo del pedido.</li> <li>• Id del pedido.</li> <li>• Plato incluido en el pedido.</li> <li>• Fecha inicial y/o final del pedido.</li> </ul>		

Tabla 22 – Requisito de usuario UR\_MC\_07

### 3.1.1.2 Requisitos de Restricción

Los requisitos de restricción, indican qué limitaciones tiene la aplicación. A continuación los vemos detallados para las aplicaciones cliente del restaurante y manager:

#### ❖ Cliente del Restaurante

IMPOSIBILIDAD DE SALIR			UR_CR_01
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario no debe poder salir de la aplicación de manera inesperada.		

Tabla 23 – Requisito de usuario UR\_CR\_01

IMPOSIBILIDAD DE ELIMINAR PLATO			UR_CR_02
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Media
Descripción	El usuario no puede eliminar un plato del pedido si el pedido está en cocina.		

Tabla 24 – Requisito de usuario UR\_CR\_02

IMPOSIBILIDAD DE VALORAR PLATO			UR_CR_03
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Media
Descripción	El usuario no puede dejar una valoración de un plato si dicho plato no ha sido pedido a la cocina durante la sesión actual.		

Tabla 25 – Requisito de usuario UR\_CR\_03

IMPOSIBILIDAD DE MODIFICAR INGREDIENTES Y/O AÑADIR COMENTARIO A LA COCINA			UR_CR_04
Prioridad	Media	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Media
Descripción	El usuario no puede modificar ingredientes o añadir un comentario para la cocina sobre un plato, después de que dicho plato haya sido añadido al pedido.		

Tabla 26 – Requisito de usuario UR\_CR\_04

## ❖ Manager

IMPOSIBILIDAD DE TERMINAR PEDIDO			UR_MR_01
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario no debe tener la opción de terminar un pedido hasta que todos los platos hayan sido marcados y se haya realizado el pago.		

Tabla 27 – Requisito de usuario UR\_MR\_01

IMPOSIBILIDAD DE MARCAR COMO PAGADO			UR_MR_02
Prioridad	Media	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	El usuario no debe tener la opción de marcar un pedido como pagado si el cliente no ha pedido la cuenta.		

Tabla 28 – Requisito de usuario UR\_MR\_02

### 3.1.2 Casos de Uso

Este apartado recoge el conjunto de situaciones en las que puede encontrarse el usuario mientras utiliza la aplicación. Para reflejarlo, utilizaremos un actor que representará al usuario y veremos, a través de un diagrama, cómo interactúa con las distintas actividades.

Para facilitar la lectura del diagrama, organizaremos la información en tablas cuyo formato incluirá los siguientes campos:

- **Nombre:** Nombre identificativo del caso de uso. Ha de ser unívoco y se utilizará a modo de resumen del mismo. No tiene que incluir detalles, pero sí ser descriptivo.
- **ID:** identificador unívoco del requisito de usuario. Su nomenclatura ha de seguir el formato: UC\_X\_YY, donde:
  - X será sustituido por 'M' en caso de tratarse de un caso de uso de la aplicación del manager, o 'C' si se trata de un caso de uso de la aplicación que maneja el cliente del restaurante desde la mesa.
  - YY será sustituido por el número de caso de uso. Comenzará en 01 e incrementará en una unidad a nuevo caso de uso.
- **Actor:** Agente externo que interactúa con el sistema.
- **Objetivo:** Acción que se persigue tras la interacción del actor con el sistema.
- **Pre-condición:** Condición que debe cumplirse *a priori* para hacer el caso de uso.
- **Escenario:** Descripción detallada de los pasos que ha de realizar el actor dentro del escenario para completar el caso de uso.
- **Escenario Alternativo:** Descripción detallada de los pasos que ha de realizar el actor dentro del escenario, en caso de que pueda darse una bifurcación en la ejecución normal del sistema.
- **Post-condición:** Hecho que se cumple si el flujo de eventos del *Escenario* se ejecuta correctamente.



## ❖ Cliente del Restaurante

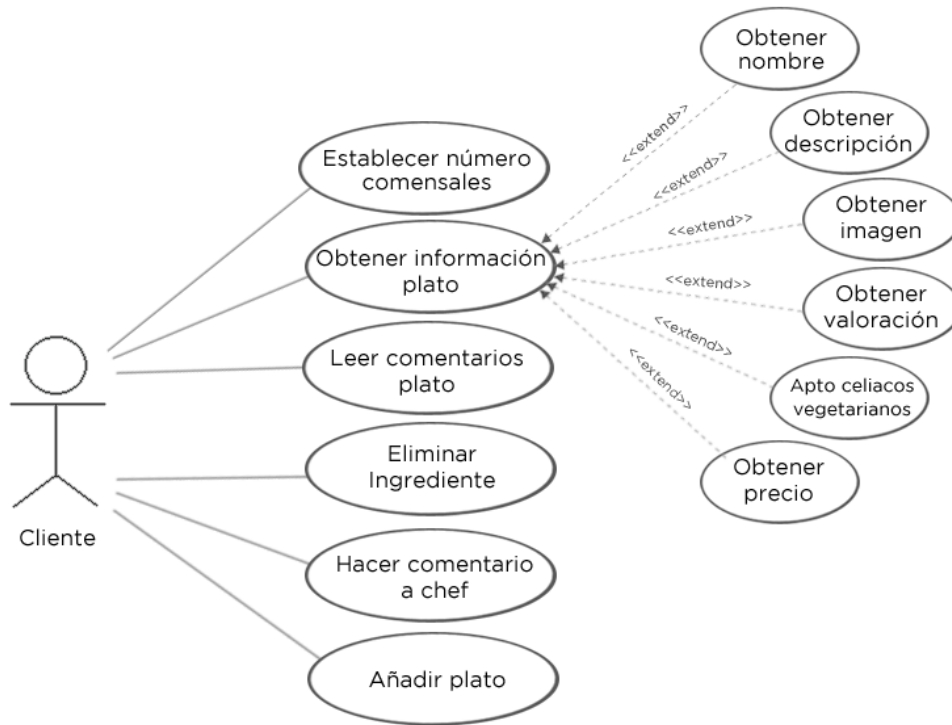


Figura 11 – Casos de Uso Cliente. Interfaz: carta del restaurante.

ESTABLECER NÚMERO COMENSALES		UC_C_01
Actor	Usuario de la aplicación cliente.	
Objetivo	Establecer el número de comensales de la mesa.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe estar configurada la conexión a base de datos.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Iniciar la aplicación.</li> <li>2. Tocar el selector de número y mantenerlo pulsado.</li> <li>3. Desplazar hacia arriba o hacia abajo hasta que la selección coincida con el número de comensales.</li> <li>4. Presionar el botón <i>Continuar</i>.</li> </ol>	
Escenario Alternativo	2b. Utilizar las flechas para seleccionar el número.	
Post-condición	Se asocia el número de comensales al pedido.	

Tabla 29 – Caso de uso UC\_C\_01

OBTENER INFORMACIÓN PLATO		UC_C_02
Actor	Usuario de la aplicación cliente.	
Objetivo	<p>Acceder a la siguiente información sobre el plato:</p> <ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Descripción.</li> <li>• Imagen.</li> <li>• Valoración y comentarios.</li> <li>• Apto para vegetarianos y celíacos.</li> <li>• Precio.</li> </ul>	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• Haber elegido una categoría y un plato del listado.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Seleccionar una categoría.</li> <li>2. Seleccionar un plato de la categoría.</li> <li>3. Seleccionar la pestaña <i>Info</i>.</li> </ol>	
Escenario Alternativo	<ol style="list-style-type: none"> <li>2b. Si se presiona el icono de volver atrás, pedirá confirmación para salir.</li> <li>3b. Pulsar el botón <i>Cancelar</i>.</li> </ol>	
Post-condición	Se muestra el nombre, la descripción, la imagen, la valoración, el precio, si es apto para celíacos y/o vegetarianos y los comentarios de otros clientes sobre el plato seleccionado.	

Tabla 30 – Caso de uso UC\_C\_02

LEER COMENTARIOS PLATO		UC_C_03
Actor	Usuario de la aplicación cliente.	
Objetivo	Leer los comentarios de un plato, hechos por clientes que han probado ese plato.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• Haber elegido una categoría y un plato del listado.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Seleccionar un plato.</li> <li>2. Seleccionar la pestaña <i>Info</i>.</li> <li>3. Pulsar el texto del comentario para cambiar a un nuevo comentario.</li> </ol>	
Escenario Alternativo	<ol style="list-style-type: none"> <li>2b. Si se presiona el icono de volver atrás, pedirá confirmación para salir.</li> <li>3b. Pulsar el botón <i>Cancelar</i>.</li> </ol>	
Post-condición	Se muestra un comentario por pantalla, navegando entre comentarios a medida que el usuario pulsa el texto del comentario actualmente mostrado.	

Tabla 31 – Caso de uso UC\_C\_03

ELIMINAR INGREDIENTE		UC_C_04
Actor	Usuario de la aplicación cliente.	
Objetivo	Eliminar ingredientes de un plato.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• Haber elegido una categoría y un plato del listado.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Seleccionar un plato.</li> <li>2. Seleccionar la pestaña <i>Opciones</i>.</li> <li>3. Pulsar el botón <i>Eliminar Ingrediente</i>.</li> <li>4. Marcar los ingredientes que se desea eliminar.</li> <li>5. Pulsar el botón <i>Hecho</i>.</li> <li>6. Pulsar el botón <i>Añadir</i>.</li> </ol>	
Escenario Alternativo	<ol style="list-style-type: none"> <li>2b. Si se presiona el icono de volver atrás, pedirá confirmación para salir.</li> <li>3b. Pulsar el botón <i>Cancelar</i>.</li> <li>5c. Si se presiona el botón atrás, muestra un mensaje de advertencia.</li> <li>6c. Pulsar el botón <i>Cancelar</i>.</li> </ol>	
Post-condición	Se indican junto al nombre los ingredientes a eliminar del plato añadido.	

Tabla 32 – Caso de uso UC\_C\_04

HACER COMENTARIO A CHEF		UC_C_05
Actor	Usuario de la aplicación cliente.	
Objetivo	Incluir un comentario al pedir un plato, para que el chef lo tenga en cuenta a la hora de cocinar el plato de dicho pedido.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• No haber enviado el pedido.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Seleccionar un plato.</li> <li>2. Seleccionar la pestaña <i>Opciones</i>.</li> <li>3. Seleccionar el botón <i>Comentario para el Chef</i></li> <li>4. Introducir el texto con el comentario que se quiera realizar.</li> <li>5. Pulsar el botón <i>Añadir</i>.</li> </ol>	
Escenario Alternativo	<ol style="list-style-type: none"> <li>2b. Si se presiona el icono de volver atrás, pedirá confirmación para salir.</li> <li>3b. Pulsar el botón <i>Cancelar</i>.</li> </ol>	
Post-condición	Se añade un comentario para el chef al plato actualmente seleccionado, que se enviará a la cocina una vez se haga el pedido.	

Tabla 33 – Caso de uso UC\_C\_05

AÑADIR PLATO		UC_C_06
Actor	Usuario de la aplicación cliente.	
Objetivo	Añadir un plato al pedido actual.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Seleccionar un plato de la carta.</li> <li>2. Presionar el botón <i>Añadir</i>.</li> </ol>	
Escenario Alternativo	2b. Si se desea hacer un comentario al chef sobre ese plato, o eliminar algún ingrediente, antes de pulsar añadir deberemos llevar dichas acciones a cabo.	
Post-condición	Se añade el plato al pedido, tantas veces como el botón <i>Añadir</i> haya sido pulsado.	

Tabla 34 – Caso de uso UC\_C\_06

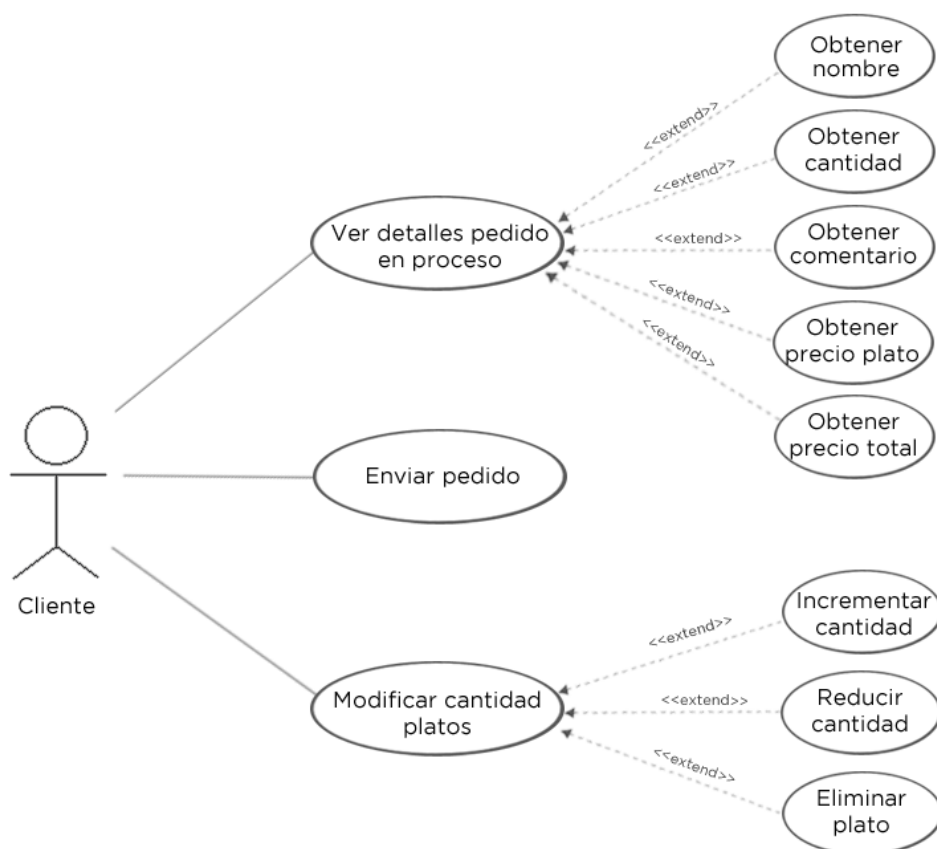


Figura 12 – Casos de Uso Cliente. Interfaz: pedido en curso (sin enviar a cocina).

VER DETALLES PEDIDO EN CURSO		UC_C_07
Actor	Usuario de la aplicación cliente.	
Objetivo	<p>Sobre un pedido que el cliente aun no haya enviado a cocina, ver la siguiente información:</p> <ul style="list-style-type: none"> <li>• Nombre de cada plato.</li> <li>• Cantidad pedida.</li> <li>• Comentarios para el cocinero (en caso de existir) por plato.</li> <li>• Precio del plato.</li> <li>• Precio total del pedido por enviar.</li> </ul>	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• Haber añadido al menos un plato al pedido.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Ver Pedido</i>.</li> <li>2. Seleccionar la pestaña <i>Pedido en Proceso</i>.</li> <li>3. Consultar la información deseada.</li> </ol>	
Escenario Alternativo	No aplica.	
Post-condición	No aplica.	

Tabla 35 – Caso de uso UC\_C\_07

MODIFICAR CANTIDAD PLATOS		UC_C_08
Actor	Usuario de la aplicación cliente.	
Objetivo	Eliminar, incrementar o reducir la cantidad que se quiere pedir de un determinado plato.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• Haber añadido un plato y no haber realizado el pedido.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Ver Pedido</i>.</li> <li>2. Seleccionar la pestaña <i>Pedido en Curso</i>.</li> <li>3. Junto a un plato, presionar el botón con el símbolo “+” si se quiere incrementar o “-” si se quiere reducir la cantidad de unidades deseadas.</li> <li>4. Para eliminar un plato, presionar el botón con el símbolo “-” del plato que queremos eliminar hasta llegar a 0 unidades.</li> </ol>	
Escenario Alternativo	No aplica.	
Post-condición	Se elimina, incrementa o reduce la cantidad de unidades a pedir de dicho plato.	

Tabla 36 – Caso de uso UC\_C\_08

ENVIAR PEDIDO		UC_C_09
Actor	Usuario de la aplicación cliente.	
Objetivo	Enviar el pedido a la cocina para comenzar su preparación.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• Debe haber al menos un plato en espera de ser enviado.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Ver Pedido</i>.</li> <li>2. Presionar la pestaña <i>Pedido en Curso</i>.</li> <li>3. Presionar el botón <i>Pedir</i>.</li> <li>4. Seleccionar <i>Sí</i> en el mensaje emergente de confirmación.</li> </ol>	
Escenario Alternativo	No Aplica	
Post-condición	Se envía el pedido al servidor.	

Tabla 37 – Caso de uso UC\_C\_09

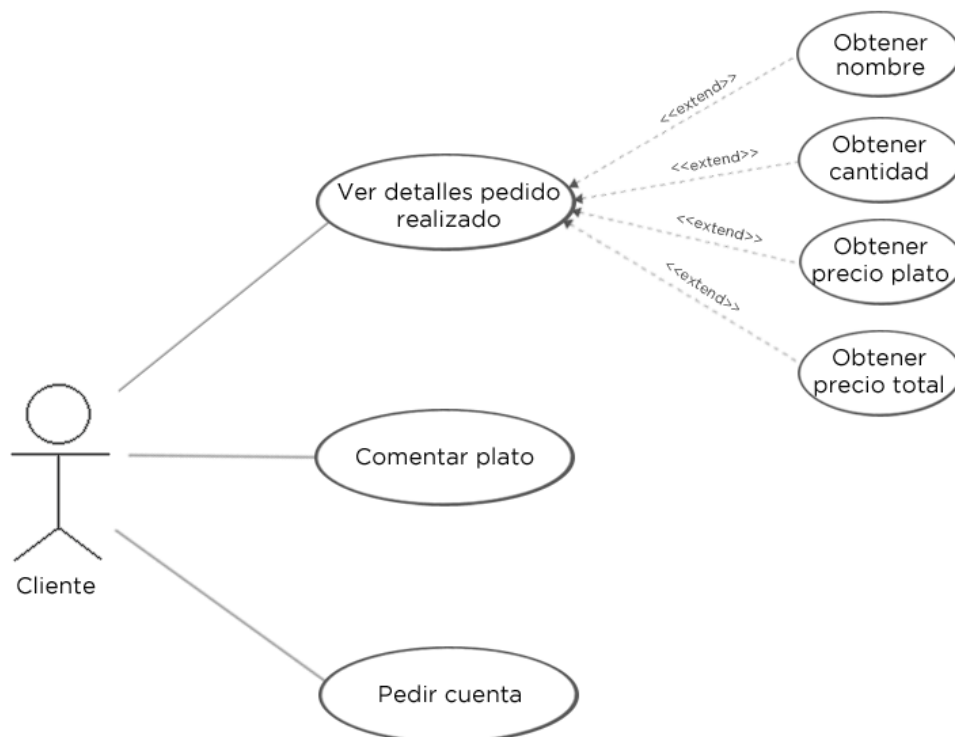


Figura 13 – Casos de Uso Cliente. Interfaz: pedido realizado (enviado a cocina).

VER DETALLES PEDIDO REALIZADO		UC_C_10
Actor	Usuario de la aplicación cliente.	
Objetivo	<p>Sobre un pedido que el cliente acabe de enviar a cocina, ver la siguiente información:</p> <ul style="list-style-type: none"> <li>• Nombre de cada plato.</li> <li>• Cantidad pedida.</li> <li>• Precio del plato.</li> <li>• Precio total del pedido realizado.</li> </ul>	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• Haber realizado el pedido.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Ver Pedido</i>.</li> <li>2. Seleccionar la pestaña <i>Pedido Realizado</i>.</li> <li>3. Consultar la información deseada.</li> </ol>	
Escenario Alternativo	No aplica.	
Post-condición	No aplica.	

Tabla 38 – Caso de uso UC\_C\_10

COMENTAR PLATO		UC_C_11
Actor	Usuario de la aplicación cliente.	
Objetivo	Crear un comentario sobre un plato pedido, para que el resto de clientes pueda verlo.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• Debe haberse realizado un pedido incluyendo el plato sobre el que se quiere comentar.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Ver Pedido</i>.</li> <li>2. Seleccionar la pestaña <i>Pedido Realizado</i>.</li> <li>3. Presionar el icono de hacer comentario junto al nombre del plato que se quiere comentar.</li> <li>4. Introducir el texto deseado en el cuadro y valorar con entre 1 y 5 estrellas (siendo 1 lo mínimo y 5 lo máximo) el plato.</li> <li>5. Pulsar el botón <i>Aceptar</i>.</li> </ol>	
Escenario Alternativo	<ol style="list-style-type: none"> <li>5b. Si se presiona el botón <i>Borrar</i>, se elimina el comentario y se oculta la opción de hacer un comentario sobre el plato.</li> <li>5c. Si se deja el campo de comentario vacío, o no se marca ninguna estrella, pulsar el botón aceptar muestra un cuadro de diálogo.</li> <li>6c. Pulsar el botón aceptar en el cuadro de diálogo.</li> </ol>	
Post-condición	Se añade un comentario sobre el plato seleccionado, que se mostrará en la carta a nuevos clientes al seleccionar el plato sobre el que se ha escrito.	

Tabla 39 – Caso de uso UC\_C\_11

PEDIR CUENTA		UC_C_12
Actor	Usuario de la aplicación cliente.	
Objetivo	Pedir el importe final del pedido para ser cobrado.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe haber conexión con la base de datos.</li> <li>• Debe haberse enviado un pedido a la cocina.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Ver Pedido</i>.</li> <li>2. Seleccionar la pestaña <i>Pedido Realizado</i>.</li> <li>3. Presionar el botón <i>Pagar</i>.</li> <li>4. Presionar la opción <i>Sí, por favor</i> en el diálogo emergente.</li> </ol>	
Escenario Alternativo	4b. Presionar la opción <i>No, aún no</i> .	
Post-condición	Se envía la información de pago al servidor.	

Tabla 40 – Caso de uso UC\_C\_12

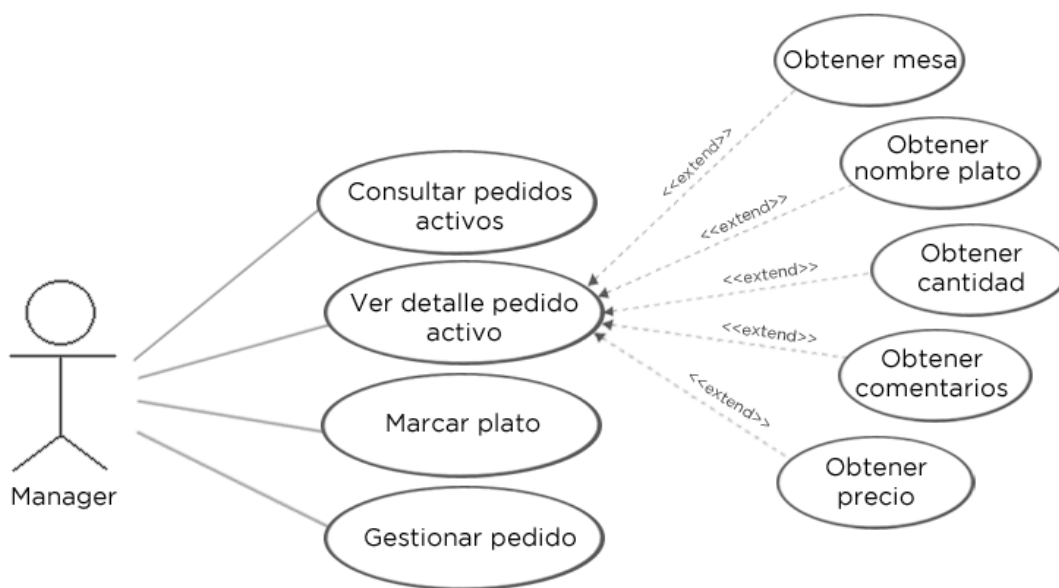


Figura 14 – Casos de Uso Manager. Interfaz: pedidos activos.



CONSULTAR PEDIDOS ACTIVOS		UC_M_01
Actor	Usuario de la aplicación manager.	
Objetivo	Ver todos los pedidos actualmente activos en el restaurante.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe estar configurada la conexión a base de datos.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Pedidos en curso</i>.</li> <li>2. Seleccionar del listado de pedidos aquel que se quiera ver.</li> </ol>	
Escenario Alternativo	No aplica.	
Post-condición	Se muestra el listado de pedidos activos y los detalles de un pedido, una vez se ha seleccionado uno de la lista.	

Tabla 41 – Caso de uso UC\_M\_01

VER DETALLES PEDIDO ACTIVO		UC_M_02
Actor	Usuario de la aplicación manager.	
Objetivo	<p>Sobre un pedido hecho por un cliente en el restaurante que aun no haya sido finalizado, obtener:</p> <ul style="list-style-type: none"> <li>• Mesa desde donde se hace el pedido.</li> <li>• Nombre de cada plato.</li> <li>• Cantidad pedida.</li> <li>• Comentarios para el cocinero (en caso de existir) por plato.</li> <li>• Precio total del pedido.</li> </ul>	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe estar configurada la conexión a base de datos.</li> <li>• Debe haber al menos un pedido activo.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Pedidos en curso</i>.</li> <li>2. Seleccionar un pedido.</li> <li>3. Consultar la información deseada.</li> </ol>	
Escenario Alternativo	No aplica.	
Post-condición	No aplica.	

Tabla 42 – Caso de uso UC\_M\_02

MARCAR PLATO		UC_M_03
Actor	Usuario de la aplicación manager.	
Objetivo	Marcar un plato de un pedido activo.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe estar configurada la conexión a base de datos.</li> <li>• Debe existir un pedido activo.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Pedidos en curso</i>.</li> <li>2. Presionar la casilla junto al plato que se desee marcar.</li> </ol>	
Escenario Alternativo	2b. Si se presiona la casilla cuando ya está marcado, se desmarca.	
Post-condición	<ul style="list-style-type: none"> <li>• Se marca la casilla junto al plato.</li> </ul>	

Tabla 43 – Caso de uso UC\_M\_03

GESTIONAR PEDIDO		UC_M_04
Actor	Usuario de la aplicación manager.	
Objetivo	Establecer el pedido como pagado y finalizarlo.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe estar configurada la conexión a base de datos.</li> <li>• Debe existir un pedido activo.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Pedidos en curso</i>.</li> <li>2. Seleccionar del listado de pedidos aquel que se finalizar.</li> <li>3. Si el cliente ha pedido y pagado la cuenta, pulsar el botón <i>Marcar Pagado</i>.</li> <li>4. Una vez todos los platos hayan sido marcados como hechos y se haya pagado el pedido, pulsar el botón <i>Finalizar Pedido</i>.</li> </ol>	
Escenario Alternativo	No aplica.	
Post-condición	<ul style="list-style-type: none"> <li>• Se elimina el pedido del listado de pedidos activos.</li> <li>• Se incluye el pedido en el listado de pedidos finalizados.</li> <li>• Se actualiza la interfaz de pedidos activos para no mostrar el recién finalizado.</li> </ul>	

Tabla 44 – Caso de uso UC\_M\_04



Figura 15 – Casos de Uso Manager. Interfaz: historial de pedidos.

CONSULTAR PEDIDOS FINALIZADOS		UC_M_05
Actor	Usuario de la aplicación manager.	
Objetivo	Ver todos los pedidos finalizados.	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe estar configurada la conexión a base de datos.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Historial de Pedidos</i>.</li> <li>2. Introducir los filtros que se deseen.</li> <li>3. Pulsar el botón <i>Buscar</i>.</li> <li>4. Seleccionar del listado de pedidos aquel que se quiera ver.</li> </ol>	
Escenario Alternativo	2b. No introducir ningún filtro.	
Post-condición	Se muestra el listado completo de pedidos finalizados y los detalles de un pedido, teniendo en cuenta los filtros aplicados.	

Tabla 45 – Caso de uso UC\_M\_05

VER DETALLES PEDIDO FINALIZADO		UC_M_06
Actor	Usuario de la aplicación manager.	
Objetivo	<p>Sobre un pedido hecho por un cliente en el restaurante que haya sido finalizado, obtener:</p> <ul style="list-style-type: none"> <li>• Mesa desde donde se hizo el pedido.</li> <li>• Fecha del pedido.</li> <li>• Nombre de cada plato.</li> <li>• Cantidad pedida.</li> <li>• Comentarios para el cocinero.</li> <li>• Precio total del pedido.</li> <li>• Si fue pagado.</li> </ul>	
Pre-condición	<ul style="list-style-type: none"> <li>• Debe existir conexión a Internet.</li> <li>• Debe estar configurada la conexión a base de datos.</li> <li>• Debe haber al menos un pedido finalizado.</li> </ul>	
Escenario	<ol style="list-style-type: none"> <li>1. Presionar el botón <i>Historial de Pedidos</i>.</li> <li>2. Seleccionar un pedido.</li> <li>3. Consultar la información deseada.</li> </ol>	
Escenario Alternativo	No aplica.	
Post-condición	No aplica.	

Tabla 46 – Caso de uso UC\_M\_06

### 3.1.3 Requisitos Software

Este apartado recoge los requisitos software del proyecto, que de nuevo se dividirán en dos categorías: funcionales y no funcionales.

Para facilitar la lectura, los requisitos serán recogidos en tablas cuyo formato incluirá los siguientes campos:

- **Nombre:** Nombre identificativo del requisito. Ha de ser unívoco y se utilizará a modo de resumen de dicho requisito. No tiene que incluir detalles, pero sí ser descriptivo.
- **ID:** identificador unívoco del requisito de usuario. Su nomenclatura ha de seguir el formato: SR\_XXY\_ZZ, donde:

- XX será sustituido por RF en caso de tratarse de un requisito funcional, o NF si no lo fuera.
  - Y será sustituido por C en caso de tratarse de un requisito de usuario de la aplicación cliente del restaurante, o M si se tratara de un requisito de la aplicación que maneja el manager.
  - ZZ será sustituido por el número de requisito dentro de su categoría. Comenzará en 01 e incrementará a cada nuevo requisito en una unidad.
- **Descripción:** Descripción detallada del requisito. Todos los detalles han de incluirse aquí para no dar lugar a ambigüedades.
  - **Prioridad:** Indica la prioridad de implementación del requisito. Facilita la organización y carga de trabajo a llevar durante el desarrollo del proyecto. Puede tomar los valores *Baja*, *Media* y *Alta*.
  - **Necesidad:** Indica como de necesaria es la introducción del requisito en el proyecto. Puede tomar los valores *Esencial* y *Deseable*.
  - **Estabilidad:** Indica si el requisito puede ser objeto de modificaciones durante el ciclo de vida del proyecto, o si por el contrario se trata de un requisito que no ha de sufrir variaciones. El campo puede tomar los valores *Estable* e *Inestable*, dependiendo de si se da el segundo o primer caso, respectivamente.
  - **Verificabilidad:** Establece con qué facilidad se puede comprobar que el requisito haya sido introducido en el proyecto. Los valores que puede tomar son *Alta*, *Media* y *Baja*.
  - **Dependencias UR:** Indica qué requisito de usuario se evalúa o implementa con dicho requisito software. Cada requisito de usuario tiene que estar cubierto por uno o varios requisitos software.

## ❖ Cliente del Restaurante

VISTA NÚMERO DE COMENSALES			SR_NFC_01
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	La primera interfaz que encuentre un usuario de la aplicación cliente de restaurante, debe mostrar un selector de número de comensales. <ul style="list-style-type: none"> <li>• El valor mínimo del selector será 1 (marcado por omisión).</li> <li>• El valor máximo del selector será 10.</li> </ul>		
Dependencias UR	UR_CC_01		

Tabla 47 – Requisito de usuario SR\_NFC\_01

VISTA Y SELECCIÓN DE PLATO			SR_NFC_02
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	La interfaz que muestra la carta del restaurante debe presentar: <ul style="list-style-type: none"> <li>• En la parte superior y alineados horizontalmente, un conjunto de pestañas que permitan al usuario cliente seleccionar una categoría de comidas.</li> <li>• En el lateral izquierdo y bajo las pestañas de categoría, un listado de todos los platos pertenecientes a la categoría seleccionada.</li> <li>• En el lateral derecho, bajo las pestañas de categoría y junto al listado de platos, los detalles del plato seleccionado.</li> </ul> Solo debe poder seleccionarse una categoría y plato a la vez.		
Dependencias UR	UR_CC_02		

Tabla 48 – Requisito de usuario SR\_NFC\_02

INFORMACIÓN Y OPCIONES PLATO			SR_NFC_03
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	Una vez elegido un plato, en el lateral derecho se mostrarán dos pestañas alineadas verticalmente, que serán: <ul style="list-style-type: none"> <li>• <b>Info:</b> al seleccionarla muestra los detalles del plato.</li> <li>• <b>Opciones:</b> al seleccionarla muestra los botones con las acciones posibles a llevar a cabo sobre el plato.</li> </ul>		
Dependencias UR	UR_CC_02, UR_CC_03, UR_CC_04		

Tabla 49 – Requisito de usuario SR\_NFC\_03

VISTA DETALLE GENERAL PLATO			SR_NFC_04
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	En el espacio bajo categorías y a la derecha del listado de platos, siempre debe mostrarse la siguiente información por pantalla: <ul style="list-style-type: none"> <li>• Imagen del plato.</li> <li>• Nombre del plato.</li> <li>• Precio del plato.</li> </ul>		
Dependencias UR	UR_CC_02		

Tabla 50 – Requisito de usuario SR\_NFC\_04

VISTA INFO PLATO			SR_NFC_05
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	Al seleccionar la pestaña <i>Info</i> , además de la información del plato que siempre debe aparecer, se añade la siguiente información: <ul style="list-style-type: none"> <li>• Descripción del plato.</li> <li>• Valoración en estrellas hechas por otros clientes.</li> <li>• Comentarios de otros clientes.</li> <li>• Si el plato es apto para celíacos.</li> <li>• Si el plato es apto para vegetarianos.</li> </ul>		
Dependencias UR	UR_CC_02		

Tabla 51 – Requisito de usuario SR\_NFC\_05

VISTA DE VALORACIONES Y COMENTARIOS			SR_NFC_06
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	En la parte de comentarios y valoraciones de la interfaz de <i>Opciones</i> del plato, se debe mostrar: <ul style="list-style-type: none"> <li>• La media de las valoraciones hechas por los clientes sobre el plato, en una escala de 1 a 5, que será representada en forma de estrellas.</li> <li>• Los comentarios hechos sobre el plato por otros clientes. Se mostrará uno cada vez y para cambiar de comentario, se tocará el área donde se encuentra el comentario actual.</li> </ul>		
Dependencias UR	UR_CC_02, UR_CC_09		

Tabla 52 – Requisito de usuario SR\_NFC\_06

VISTA OPCIONES PLATO			SR_NFC_07
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Al seleccionar la pestaña <i>Opciones</i>, además de la información del plato que siempre debe aparecer, se muestran dos botones:</p> <ul style="list-style-type: none"> <li>• Eliminar ingrediente (solo si hay ingredientes eliminables).</li> <li>• Comentario a cocina.</li> </ul>		
Dependencias UR	UR_CC_03, UR_CC_04		

Tabla 53 – Requisito de usuario SR\_NFC\_07

BOTÓN ELIMINAR INGREDIENTE			SR_RFC_08
Prioridad	Media	Necesidad	Deseable
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Pulsando el botón <i>Eliminar Ingrediente</i> de la pestaña <i>Opciones</i>, debe mostrarse una nueva interfaz en la que aparezcan:</p> <ul style="list-style-type: none"> <li>• Todos los ingredientes que se pueden eliminar del plato.</li> <li>• Un botón para volver al menú una vez se hayan marcado los ingredientes deseados.</li> </ul>		
Dependencias UR	UR_CC_03		

Tabla 54 – Requisito de usuario SR\_RFC\_08

ACCIÓN ELIMINAR INGREDIENTE			SR_RFC_09
Prioridad	Media	Necesidad	Deseable
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>En la interfaz de eliminación de ingredientes, se marcará cada ingrediente a eliminar pulsándolo hasta que el color del ingrediente quede sombreado y aparezca un aspa roja sobre él. Se podrán eliminar tantos ingredientes como se deseen.</p>		
Dependencias UR	UR_CC_03		

Tabla 55 – Requisito de usuario SR\_RFC\_09

ACCIÓN COMENTARIO PARA COCINERO			SR_RFC_10
Prioridad	Media	Necesidad	Deseable
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Pulsando el botón <i>Comentario a cocina</i> de la pestaña <i>Opciones</i>, debe aparecer un cuadro de texto donde el usuario pueda escribir mediante el teclado software aquello que desee enviar a la cocina previa la preparación del plato.</p>		
Dependencias UR	UR_CC_04		

Tabla 56 – Requisito de usuario SR\_RFC\_10



GUARDAR OPCIONES DEL PLATO			SR_RFC_11
Prioridad	Media	Necesidad	Deseable
Estabilidad	Estable	Verificabilidad	Alta
Descripción	Una vez hayan sido marcados como eliminados los ingredientes de un plato y/o se haya introducido un comentario para el cocinero, se mantendrán guardados los valores hasta que se pulse el botón añadir o se elija un plato distinto del menú.		
Dependencias UR	UR_CC_03, UR_CC_04		

Tabla 57 – Requisito de usuario SR\_RFC\_11

BOTÓN AÑADIR PLATO A PEDIDO			SR_RFC_12
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	En la vista principal del menú, debe existir un botón que permita añadir el plato seleccionado al pedido. Al pulsarlo, debe aparecer un mensaje de notificación que avise de que ha sido añadido al pedido.		
Dependencias UR	UR_CC_05		

Tabla 58 – Requisito de usuario SR\_RFC\_12

BOTÓN VER PEDIDO			SR_RFC_13
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>En la vista principal del menú, debe existir un botón que permita al usuario ver la interfaz de pedidos. Al pulsarlo, mostrarse una interfaz nueva con dos pestañas en la parte superior, correspondientes a las acciones de:</p> <ul style="list-style-type: none"> <li>Ver pedidos aún sin enviar a cocina.</li> <li>Ver pedidos que han sido enviados a cocina.</li> </ul> <p>Por omisión, se mostrará la pestaña de pedidos aún sin enviar.</p>		
Dependencias UR	UR_CC_10		

Tabla 59 – Requisito de usuario SR\_RFC\_13

VISTA PEDIDO EN CURSO			SR_NFC_14
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Si hay al menos un plato en el pedido, en la interfaz que muestra los pedidos aún sin enviar a cocina, debe aparecer:</p> <ul style="list-style-type: none"> <li>Un listado de todos los platos pedidos, y en cada línea del listado debe aparecer: <ul style="list-style-type: none"> <li>Nombre del plato pedido y los ingredientes eliminados.</li> <li>Cantidad de dicho plato pedida.</li> <li>Botón de incrementar y reducir la cantidad.</li> <li>En caso de haberse realizado un comentario para el cocinero, un botón que muestre que hay un comentario.</li> <li>Precio del plato multiplicado por la cantidad de veces pedido.</li> </ul> </li> <li>Precio total del pedido sin enviar a cocina.</li> <li>Botón de realizar pedido.</li> <li>Botón de ver carta.</li> </ul>		
Dependencias UR	UR_CC_10		

Tabla 60 – Requisito de usuario SR\_NFC\_14

ACCIÓN MODIFICAR CANTIDAD PLATO			SR_RFC_15
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>En la interfaz de pedido sin enviar:</p> <ul style="list-style-type: none"> <li>El botón de incrementar la cantidad pedida de un plato añadirá una unidad a dicha cantidad. Puede ser pulsado tantas veces como se desee.</li> <li>El botón de reducir la cantidad pedida de un plato eliminará una unidad de dicha cantidad, siempre y cuando esta cantidad sea mayor que cero.</li> </ul>		
Dependencias UR	UR_CC_05, UR_CC_06		

Tabla 61 – Requisito de usuario SR\_RFC\_15

ACCIÓN ELIMINAR PLATO			SR_RFC_16
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Solo se podrá eliminar un plato desde la interfaz de pedidos sin enviar. Cuando la cantidad de unidades pedidas de un plato sea cero, el plato quedará eliminado del pedido.</p> <p>No se eliminará la línea de la lista del pedido hasta que se vuelva al menú o se realice el envío, ya que el cliente puede haber marcado por error el 0 y se le da la oportunidad de utilizar el botón de incrementar la cantidad del plato mientras no abandone la vista.</p>		
Dependencias UR	UR_CC_06		

Tabla 62 – Requisito de usuario SR\_RFC\_16

BOTÓN REALIZAR PEDIDO			SR_RFC_17
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Al pulsar el botón <i>Enviar Pedido</i> de la interfaz <i>Pedido en curso</i>, se pedirá confirmación del usuario para verificar que no ha pulsado el botón por error. Si confirma, el pedido y los comentarios se enviará al servidor para su almacenamiento en base de datos.</p> <p>Se notificará al usuario de que se ha recibido en cocina, y se volverá a la interfaz de la carta del restaurante.</p>		
Dependencias UR	UR_CC_07		

Tabla 63 – Requisito de usuario SR\_RFC\_17

ACCION AÑADIR PLATO TRAS ENVIAR PEDIDO			SR_RFC_18
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Una vez haya sido enviado un pedido a cocina, el usuario debe poder seguir añadiendo platos al pedido para enviarlos posteriormente. Se realizará del mismo modo que antes de enviar el pedido.</p> <p>Una vez haya sido enviado a cocina, se mostrará en la interfaz de <i>Pedido Realizado</i> el conjunto de todos los platos que se han enviado a la cocina en la sesión actual, así como la suma del precio de todos los platos.</p>		
Dependencias UR	UR_CC_05		

Tabla 64 – Requisito de usuario SR\_RFC\_18

VISTA PEDIDO ENVIADO			SR_NFC_19
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Si se ha enviado un pedido a la cocina, en la interfaz que muestra los pedidos enviados, debe aparecer:</p> <ul style="list-style-type: none"> <li>• Un listado de todos los platos pedidos, y en cada línea del listado debe aparecer: <ul style="list-style-type: none"> <li>○ Nombre del plato pedido.</li> <li>○ Cantidad de dicho plato pedida.</li> <li>○ Precio del plato multiplicado por la cantidad de veces pedido.</li> <li>○ Icono para escribir un comentario del plato.</li> </ul> </li> <li>• Precio total del pedido realizado.</li> <li>• Botón de pagar.</li> </ul>		
Dependencias UR	UR_CC_10		

Tabla 65 – Requisito de usuario SR\_NFC\_19

BOTÓN HACER COMENTARIO PLATO			SR_RFC_20
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Desde la interfaz que muestra los pedidos enviados, al pulsar sobre el botón de escribir comentario sobre el plato, debe aparecer en el lado derecho:</p> <ul style="list-style-type: none"> <li>• Nombre del plato a valorar.</li> <li>• Cuadro de texto para introducir el comentario (máx: 255 caracteres).</li> <li>• Rating para valorar entre 1 y 5 estrellas.</li> <li>• Botón de aceptar comentario.</li> <li>• Botón de borrar comentario.</li> </ul>		
Dependencias UR	UR_CC_08		

Tabla 66 – Requisito de usuario SR\_RFC\_20

ACCIÓN VALORAR PLATO PEDIDO			SR_RFC_21
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Desde la interfaz que muestra los pedidos enviados y tras pulsar el botón de escribir comentario sobre un plato:</p> <ul style="list-style-type: none"> <li>• Si ha sido introducido un comentario y se ha marcado al menos una estrella, al pulsar el botón aceptar debe enviarse el comentario al servidor.</li> <li>• Si alguno de los campos no ha sido introducido y se pulsa aceptar, aparecerá un aviso indicando que ha de introducirse tanto un comentario como una valoración en estrellas.</li> <li>• Si se pulsa el botón borrar, se eliminará el comentario sin llegar a enviarse.</li> </ul>		
Dependencias UR	UR_CC_08		

Tabla 67 – Requisito de usuario SR\_RFC\_21

ACCIÓN PEDIR CUENTA			SR_RFC_22
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Al pulsar el botón <i>Pedir cuenta</i>, debe mostrarse una notificación alertando de que no podrán pedirse más platos en la sesión actual si se lleva a cabo la acción.</p> <p>Si el cliente pulsa el botón de aceptación, se envía la orden de que la cuenta ha sido pedida al servidor y se termina la sesión, mostrando una interfaz de despedida que será desactivada por el manager.</p> <p>Si el cliente pulsa el botón de cancelar, se cierra la notificación y se puede continuar pidiendo.</p>		
Dependencias UR	UR_CC_11		

Tabla 68 – Requisito de usuario SR\_RFC\_22

NO PERMITIR SALIR DE LA APLICACIÓN			SR_RFC_23
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario no debe poder abandonar la sesión hasta haber pagado. Si utiliza el botón de volver atrás en la pantalla de la carta del restaurante, debe salir un diálogo pidiendo la contraseña de administrador para realizar la operación.		
Dependencias UR	UR_CR_01		

Tabla 69 – Requisito de usuario SR\_RFC\_23

NO PERMITIR ELIMINAR PLATO EN PEDIDO ENVIADO			SR_RFC_24
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Media
Descripción	No se da la opción de que el usuario pueda eliminar un plato del pedido si el pedido ha sido enviado a la cocina.		
Dependencias UR	UR_CR_02		

Tabla 70 – Requisito de usuario SR\_RFC\_24

NO PERMITIR VALORAR PLATO SIN ESTAR EN PEDIDO REALIZADO			SR_RFC_25
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Media
Descripción	No se permite que un usuario valore y comente un plato si este no ha sido pedido y enviado a cocina en la sesión actual.		
Dependencias UR	UR_CR_03		

Tabla 71 – Requisito de usuario SR\_RFC\_25

NO PERMITIR MODIFICAR PEDIDO ENVIADO			SR_RFC_26
Prioridad	Media	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Media
Descripción	No se permite que un usuario pueda modificar ingredientes o añadir un comentario para la cocina sobre un plato, después de que dicho plato haya sido añadido al pedido.		
Dependencias UR	UR_CR_04		

Tabla 72 – Requisito de usuario SR\_RFC\_26

## ❖ Manager

SELECCIÓN DE VISTA			SR_NFM_01
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>Deben existir dos pestañas alineadas horizontalmente y siempre visibles en la parte superior de la interfaz, que serán:</p> <ul style="list-style-type: none"> <li>• <b>Pedidos activos:</b> muestra la interfaz de todos los pedidos actualmente activos en el restaurante.</li> <li>• <b>Historial de pedidos:</b> muestra la interfaz de búsqueda y visionado de pedidos finalizados.</li> </ul>		
Dependencias UR	UR_MC_01, UR_MC_06		

Tabla 73 – Requisito de usuario SR\_NFM\_01

VISTA PEDIDOS ACTIVOS			SR_NFM_02
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	En la interfaz de pedidos activos en el restaurante, debe mostrarse un listado en la parte izquierda de todas las mesas que han enviado un pedido a cocina y aun no ha sido terminado.		
Dependencias UR	UR_MC_01		

Tabla 74 – Requisito de usuario SR\_NFM\_02

VISTA DETALLE DE PEDIDOS ACTIVOS			SR_NFM_03
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>En la interfaz de pedidos activos, debe mostrarse:</p> <ul style="list-style-type: none"> <li>• En la parte izquierda, un listado de todas las mesas que han enviado un pedido a cocina y aun no haya sido terminado.</li> <li>• A la derecha del listado, los detalles del pedido: <ul style="list-style-type: none"> <li>○ Número de mesa seleccionado.</li> <li>○ Listado de todos los platos pedidos por la mesa, donde cada línea mostrará: <ul style="list-style-type: none"> <li>▪ Checkbox.</li> <li>▪ Platos pedidos.</li> <li>▪ Cantidad de cada plato.</li> <li>▪ Comentarios para cocina junto al plato (si los hay).</li> </ul> </li> <li>○ Precio total de la mesa seleccionada.</li> <li>○ Botón de actualización de interfaz.</li> </ul> </li> </ul>		
Dependencias UR	UR_MC_01		

Tabla 75 – Requisito de usuario SR\_NFM\_03

MARCACIÓN PLATOS			SR_RFM_04
Prioridad	Media	Necesidad	Deseable
Estabilidad	Estable	Verificabilidad	Alta
Descripción	El usuario podrá pulsar el checkbox junto a cada plato para marcarlo o desmarcarlo. Deberá guardarse la información de cada plato marcado en cada mesa activa hasta que se salga de la aplicación.		
Dependencias UR	UR_MC_02		

Tabla 76 – Requisito de usuario SR\_RFM\_04

AVISO DE NOTIFICACIÓN DE CUENTA			SR_RFM_05
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	Si una mesa activa pide la cuenta desde la aplicación, aparecerá un icono en el listado de mesas activas, junto al nombre de la mesa que ha pedido la cuenta. Además, se mostrará un botón en la interfaz del manager que permitirá marcar el pedido como pagado.		
Dependencias UR	UR_MC_03		

Tabla 77 – Requisito de usuario SR\_RFM\_05

ACCIÓN MARCAR PAGADO			SR_RFM_06
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	Al pulsar el botón <i>Marcar Pagado</i> una vez el cliente ha pedido la cuenta, se enviará la información del pago final al servidor y quedará almacenada en base de datos.		
Dependencias UR	UR_MC_04		

Tabla 78 – Requisito de usuario SR\_RFM\_06

VISTA BOTÓN FINALIZAR PEDIDO			SR_NFM_07
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Media
Descripción	Si todos los platos de un pedido han sido marcados en la interfaz de pedidos activos y el pedido ha sido marcado como pagado, deberá mostrarse un botón que permita finalizar el pedido.		
Dependencias UR	UR_MC_04		

Tabla 79 – Requisito de usuario SR\_NFM\_07



ACCIÓN BOTÓN FINALIZAR PEDIDO			SR_RFM_08
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Media
Descripción	<p>Al pulsar el botón <i>Finalizar pedido</i> el pedido deja de ser un pedido activo y pasa a ser un pedido finalizado, lo que implica:</p> <ul style="list-style-type: none"> <li>• El pedido deja de mostrarse en la interfaz de pedidos activos.</li> <li>• El pedido puede mostrarse en la interfaz de historial de pedidos (dependiendo de los filtros aplicados).</li> <li>• Se envía la información de que el pedido ha sido terminado al servidor.</li> </ul>		
Dependencias UR	UR_MC_04		

Tabla 80 – Requisito de usuario SR\_RFM\_08

ACCIÓN ACTUALIZAR INTERFAZ			SR_RFM_09
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	<p>Al pulsar el botón de actualizar interfaz, se hace una llamada al servidor para recuperar todos los pedidos activos y se muestra el listado de mesas activas con todas las recuperadas en ese momento.</p>		
Dependencias UR	UR_MC_05		

Tabla 81 – Requisito de usuario SR\_RFM\_09

VISTA HISTORIAL DE PEDIDOS			SR_NFM_10
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	<p>La interfaz deberá estar dividida en dos partes:</p> <ul style="list-style-type: none"> <li>• En la parte superior se mostrarán los filtros de búsqueda.</li> <li>• En la parte inferior se mostrarán los resultados de la búsqueda. Estará organizado a su vez en dos partes: <ul style="list-style-type: none"> <li>○ En la izquierda se muestra el listado de pedidos de acuerdo con los filtros establecidos.</li> <li>○ En la derecha se muestran los detalles del pedido seleccionado, que serán: <ul style="list-style-type: none"> <li>▪ ID del pedido seleccionado.</li> <li>▪ Número de mesa desde donde se hizo el pedido.</li> <li>▪ Platos y cantidad de ellos pedidos.</li> <li>▪ Comentarios de los platos (si hubiera).</li> <li>▪ Fecha del pedido.</li> <li>▪ Precio final del pedido.</li> <li>▪ Si el pedido fue pagado.</li> </ul> </li> </ul> </li> </ul>		
Dependencias UR	UR_MC_06		

Tabla 82 – Requisito de usuario SR\_NFM\_10

VISTA DE BÚSQUEDA DE PEDIDOS			SR_NFM_11
Prioridad	Alta	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	<p>En la interfaz de historial de pedidos, deben mostrarse los siguientes filtros de búsqueda:</p> <ul style="list-style-type: none"> <li>• Número de mesa en la que se realizó el pedido.</li> <li>• Precio.</li> <li>• Id del pedido.</li> <li>• Plato incluido en el pedido.</li> <li>• Fecha inicial y final.</li> </ul>		
Dependencias UR	UR_MC_07		

Tabla 83 – Requisito de usuario SR\_NFM\_11

FILTRO DE BÚSQUEDA: Nº MESA			SR_RFM_12
Prioridad	Media	Necesidad	Deseable
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	<ul style="list-style-type: none"> <li>• Al filtrar por número de mesa, deben devolverse todos los pedidos realizados desde la mesa indicada.</li> <li>• El valor del identificador introducido desde la interfaz ha de corresponder exactamente con el del pedido devuelto.</li> <li>• Si no existe ningún pedido que se corresponda con el filtro establecido, se notificará con un aviso al usuario y no se mostrará ningún pedido.</li> </ul>		
Dependencias UR	UR_MC_07		

Tabla 84 – Requisito de usuario SR\_RFM\_12

FILTRO DE BÚSQUEDA: PRECIO			SR_RFM_13
Prioridad	Media	Necesidad	Deseable
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	<p>Al filtrar por precio, se podrá indicar un mínimo y un máximo:</p> <ul style="list-style-type: none"> <li>• Si se indica un precio mínimo, se devolverán todos los pedidos cuyo precio final sea igual o superior al indicado.</li> <li>• Si se indica un precio máximo, se devolverán todos los pedidos cuyo precio final sea igual o inferior al indicado.</li> <li>• Si el precio máximo es igual al mínimo, se devolverán los pedidos cuyo precio sea el indicado en ambos campos.</li> <li>• Si uno de los campos (máximo o mínimo) queda vacío, no existirá restricción de precio por ese lado.</li> <li>• Si el precio mínimo es mayor que el precio máximo, no se devolverán resultados.</li> </ul>		
Dependencias UR	UR_MC_07		

Tabla 85 – Requisito de usuario SR\_RFM\_13

FILTRO DE BÚSQUEDA: PLATO			SR_RFM_14
Prioridad	Media	Necesidad	Deseable
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	<ul style="list-style-type: none"> <li>Al filtrar por plato se devolverán todos aquellos pedidos que contengan el plato indicado.</li> <li>Se podrá introducir el nombre completo del plato o parte de él.</li> <li>Si no existe ningún pedido que se corresponda con el filtro establecido, se notificará con un aviso al usuario y no se mostrará ningún pedido.</li> </ul>		
Dependencias UR	UR_MC_07		

Tabla 86 – Requisito de usuario SR\_RFM\_14

FILTRO DE BÚSQUEDA: ID PEDIDO			SR_RFM_15
Prioridad	Media	Necesidad	Deseable
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	<p>Al filtrar por Id Pedido, se devolverá:</p> <ul style="list-style-type: none"> <li>El pedido cuyo id coincida con el introducido, si se detalla completamente.</li> <li>Un conjunto de pedidos cuyo identificador contiene la cadena detallada en el filtro.</li> </ul> <p>Si no existe ningún pedido que se corresponda con el filtro establecido, se notificará con un aviso al usuario y no se mostrará ningún pedido.</p>		
Dependencias UR	UR_MC_07		

Tabla 87 – Requisito de usuario SR\_RFM\_15

FILTRO DE BÚSQUEDA: FECHA			SR_RFM_16
Prioridad	Media	Necesidad	Deseable
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	<p>Al filtrar por fecha se podrá indicar una fecha inicial y una fecha final:</p> <ul style="list-style-type: none"> <li>Si se indica una fecha inicial, se devolverán todos los pedidos cuya fecha sea igual o posterior a la indicada.</li> <li>Si se indica una fecha final, se devolverán todos los pedidos cuya fecha sea igual o anterior a la indicada.</li> <li>Si ambas fechas son iguales, se devolverán los pedidos cuya fecha sea exactamente la indicada.</li> <li>Si la fecha inicial es mayor que la final, no se devolverán resultados.</li> <li>Si uno de los campos (fecha inicial o final) queda vacío, no existirá restricción de fecha por ese lado.</li> </ul>		
Dependencias UR	UR_MC_07		

Tabla 88 – Requisito de usuario SR\_RFM\_16

CONDICIONES DE TERMINAR PEDIDO			SR_RFM_17
Prioridad	Media	Necesidad	Esencial
Estabilidad	Estable	Verificabilidad	Alta
Descripción	No se permite al usuario la terminación de un pedido hasta que todos los platos hayan sido marcados y se haya realizado el pago del pedido.		
Dependencias UR	UR_MR_01		

Tabla 89 – Requisito de usuario SR\_RFM\_17

CONDICIONES DE MARCAR COMO PAGADO			SR_RFM_18
Prioridad	Media	Necesidad	Esencial
Estabilidad	Inestable	Verificabilidad	Alta
Descripción	No se permite al usuario marcar un pedido como pagado si el cliente no ha pedido la cuenta.		
Dependencias UR	UR_MR_02		

Tabla 90 – Requisito de usuario SR\_RFM\_18

### 3.1.4 Matriz de trazabilidad

Una vez recogidos y analizados tanto los requisitos de usuario como los requisitos software, ha de comprobarse que todo requisito de usuario esté contemplado por al menos un requisito software.

Para hacer esta comprobación, se utilizan matrices de trazabilidad. Una matriz de trazabilidad tiene en su eje vertical el conjunto de identificadores de requisitos de usuario, tanto de capacidad como de restricción, y en su eje horizontal, los identificadores de requisitos software.

Si recordamos del apartado anterior, los requisitos software tienen un campo llamado *Dependencias UR* que indica el requisito de usuario evaluado o implementado por el requisito software en cuestión. Siendo así, la matriz de trazabilidad marcará cada requisito de usuario contemplado por un requisito software concreto, siempre y cuando este haya sido asignado en el campo *Dependencias UR* correspondiente.

Si una línea de la matriz quedara completamente en blanco, significaría que en la aplicación final no se vería implementado el requisito correspondiente a dicha fila y el proyecto quedaría incompleto. Para comprobar que esto no ocurre, analizamos las matrices de trazabilidad para las dos aplicaciones.

	SR_NFC_01	SR_NFC_02	SR_NFC_03	SR_NFC_04	SR_NFC_05	SR_NFC_06	SR_NFC_07	SR_RFC_08	SR_RFC_09	SR_RFC_10	SR_RFC_11	SR_RFC_12	SR_RFC_13	SR_NFC_14	SR_RFC_15	SR_RFC_16	SR_RFC_17	SR_RFC_18	SR_NFC_19	SR_RFC_20	SR_RFC_21	SR_RFC_22	SR_RFC_23	SR_RFC_24	SR_RFC_25	SR_RFC_26
UR_CC_01	X																									
UR_CC_02		X	X	X	X	X																				
UR_CC_03			X				X	X	X		X															
UR_CC_04			X				X			X	X															
UR_CC_05												X			X			X								
UR_CC_06															X	X										
UR_CC_07																	X									
UR_CC_08																			X	X						
UR_CC_09						X																				
UR_CC_10													X	X					X							
UR_CC_11																						X				
UR_CR_01																							X			
UR_CR_02																								X		
UR_CR_03																									X	
UR_CR_04																										X

Tabla 91 – Matriz de trazabilidad de requisitos del cliente de restaurante.

	SR_NFM_01	SR_NFM_02	SR_NFM_03	SR_RFM_04	SR_RFM_05	SR_RFM_06	SR_NFM_07	SR_RFM_08	SR_RFM_09	SR_NFM_10	SR_NFM_11	SR_RFM_12	SR_RFM_13	SR_RFM_14	SR_RFM_15	SR_RFM_16	SR_RFM_17	SR_RFM_18
UR_MC_01	X	X	X															
UR_MC_02				X														
UR_MC_03					X													
UR_MC_04						X	X											
UR_MC_05								X										
UR_MC_06	X								X									
UR_MC_07										X	X	X	X	X	X	X		
UR_MR_01																	X	
UR_MR_02																		X

Tabla 92 – Matriz de trazabilidad de requisitos del manager.

En ambas matrices todos los requisitos de usuario quedan cubiertos por al menos un requisito software, lo que lleva a afirmar que todos los requisitos establecidos por el usuario serán implementados y por tanto no es necesario hacer un nuevo estudio de requisitos software.

## 3.2 Diseño

En este apartado se estudiarán los elementos relacionados con la fase de diseño del ciclo de desarrollo software. Se analizará el diseño arquitectónico del sistema, las interfaces con las que el usuario interactuará y la estructura de base de datos con la que trabajará el servidor.

### 3.2.1 Diseño de la Arquitectura

El sistema se basa en una arquitectura Cliente-Servidor, que se trata de una arquitectura distribuida en la que una o varias aplicaciones cliente hacen peticiones a un servidor, que a su vez da respuesta recuperando y devolviendo información a los clientes que hicieron la petición.

En el proyecto actual, se cuenta con dos aplicaciones – cliente y manager – que envían peticiones al servidor con el fin de introducir o recuperar información de base de datos. Será el servidor el encargado de, una vez recibida una petición, hacer la consulta oportuna a base de datos y enviar la información obtenida a los clientes.

Existirán distintos tipos de peticiones entre los clientes y el servidor, pudiendo ser de recuperación de información o de inserción, pero todas las comunicaciones se realizan partiendo de una petición del cliente a la que el servidor ha de responder, y nunca al contrario.

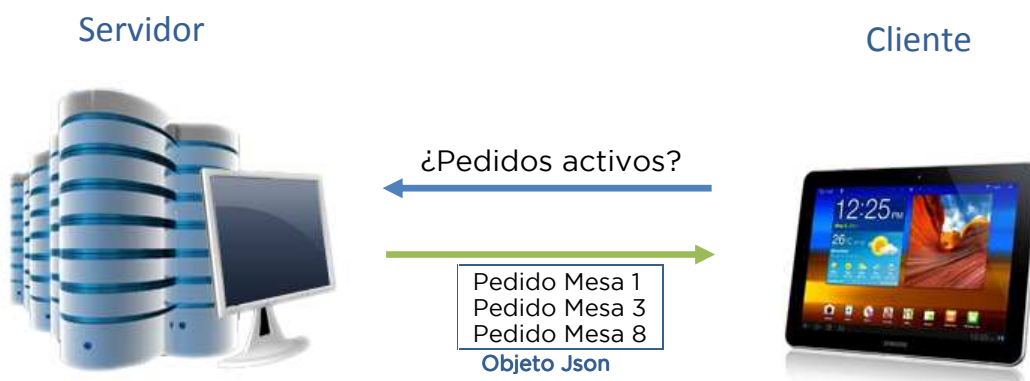


Figura 16 – Ejemplo de comunicación en el sistema.

Se utilizará un diseño Cliente-Servidor en 3 niveles, donde:

- **Nivel 1.** Corresponde al cliente, es decir, el dispositivo que envía una solicitud y que dispone de una interfaz a través de la cual el cliente puede interactuar con el sistema.
- **Nivel 2.** Se trata del servidor web, cuya tarea es proporcionar los recursos solicitados o realizar las operaciones indicadas.
- **Nivel 3.** El servidor de datos que proporciona al servidor de aplicaciones los datos requeridos.

El esquema que representa a esta arquitectura Cliente-Servidor de 3 niveles, es la siguiente.

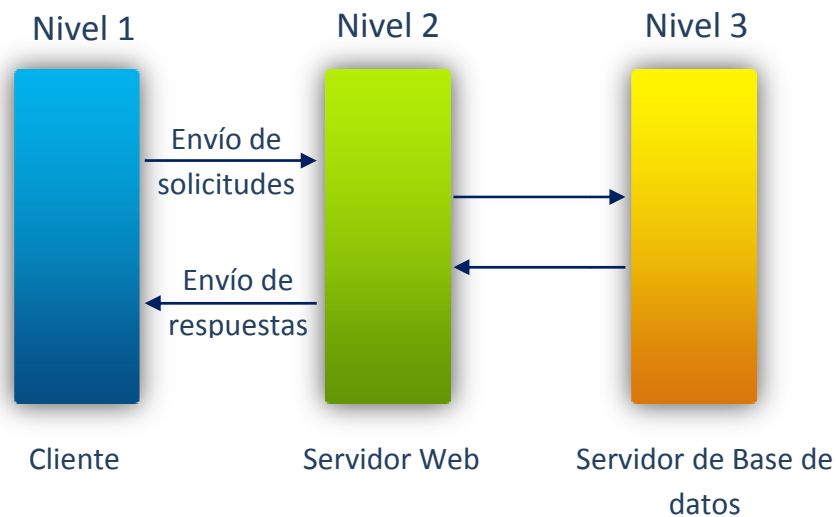


Figura 17 – Arquitectura Cliente-Servidor de 3 niveles

De esta manera, el sistema cuenta con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelo el negocio) y otra para el acceso a datos.

Actualmente, en el desarrollo de software es imprescindible la utilización de patrones de diseño que otorguen a las aplicaciones características que les permitan ser fácilmente flexibles, evolutivas y de fácil mantenimiento.

Este proyecto se desarrolla utilizando el patrón Modelo-Vista-Controlador (MVC), el cual se adapta perfectamente a la arquitectura Cliente-Servidor y permite hacer una separación entre la parte gráfica y los procesos lógicos y de datos de las aplicaciones, de acuerdo a la arquitectura de tres niveles existente.

En el patrón MVC existen tres elementos:

- **Modelo:** es el conjunto de clases que representan la información del mundo real que el sistema debe reflejar.
- **Vista:** es la encargada de la representación visual del modelo al usuario.
- **Controlador:** recibe, trata y responde a los eventos enviados por el usuario. Interactúa tanto con el modelo como con la vista.

El flujo de control del MVC donde se detalla paso a paso como se realizarían las llamadas y comunicaciones en la aplicación, es el siguiente:

1. Se realiza una petición desde la aplicación.
2. El controlador recibe la petición y la gestiona.
3. El controlador consulta o modifica el modelo.
4. El controlador envía la respuesta a la vista.
5. La vista reacciona a la respuesta modificando la interfaz.

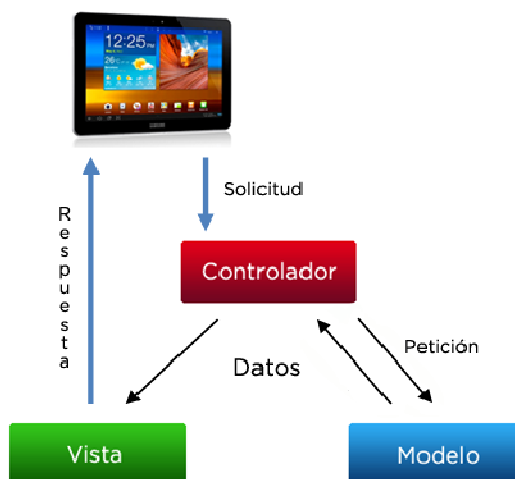


Figura 18 – Flujo de control del MVC en la aplicación.



Una vez conocida la arquitectura y el patrón de diseño, es necesario analizar los módulos en que se dividirá cada aplicación para poder diseñar el diagrama de componentes.

En la siguiente figura, se representan los componentes de las aplicaciones cliente y manager del sistema, así como de servidor y base de datos.

Las relaciones entre componentes están representadas mediante flechas, y la dirección que toman dichas flechas corresponde al flujo de datos existente entre los componentes que relacionan.

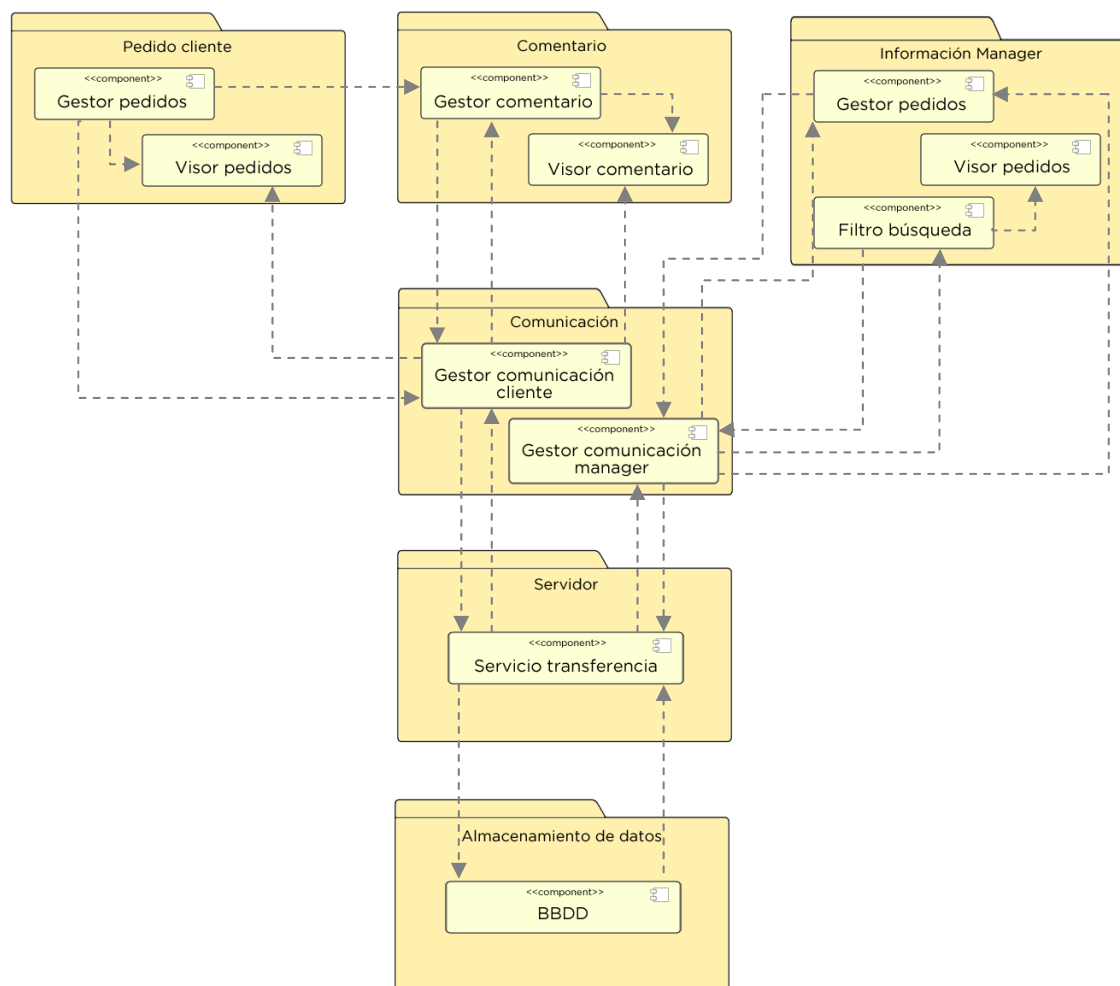


Figura 19 – Diagrama de componentes.

A continuación son especificados todos los componentes en detalle. Sobre cada componente nos interesa conocer:

- **Nombre:** nombre identificativo del componente.
- **ID:** identificador unívoco del componente. Su nomenclatura ha de seguir el formato: COM\_XX, donde XX será sustituido por un número comenzando en 1.
- **Propósito:** finalidad que persigue el componente en la aplicación.
- **Función:** listado de funcionalidades que llevará a cabo el componente en el sistema.
- **Referencias:** indica qué requisitos de usuario cubre este componente. Es importante comprobar que todos los requisitos queden cubiertos por los componentes.

GESTOR DE PEDIDOS CLIENTE		COM_01
Propósito	Establecer la configuración inicial de sesión y mantener un registro de los platos añadidos al pedido durante la sesión.	
Función	<ul style="list-style-type: none"> <li>• Establecer el número de comensales.</li> <li>• Añadir plato.</li> <li>• Eliminar plato.</li> <li>• Incrementar cantidad de platos iguales.</li> <li>• Eliminar ingredientes del plato.</li> </ul>	
Referencias	UR_CC_01, UR_CC_03, UR_CC_05, UR_CC_06, UR_CR_01, UR_CR_04	

Tabla 93 – Componente Gestor de pedidos de cliente

VISUALIZADOR DE PEDIDOS CLIENTE		COM_02
Propósito	Mostrar el estado del pedido actual en la aplicación del cliente de restaurante.	
Función	<ul style="list-style-type: none"> <li>• Mostrar la carta sobre la que se hará el pedido.</li> <li>• Mostrar el pedido por enviar a la cocina.</li> <li>• Mostrar el pedido ya enviado a la cocina.</li> <li>• Mostrar el precio de los platos individualmente.</li> <li>• Mostrar la suma de precios de los platos pedidos.</li> </ul>	
Referencias	UR_CC_02, UR_CC_10, UR_CR_01, UR_CR_02, UR_CR_03	

Tabla 94 – Componente Visualizador de pedidos de cliente

GESTOR DE COMENTARIOS		COM_03
Propósito	Mantener un registro de comentarios introducido por el cliente del restaurante.	
Función	<ul style="list-style-type: none"> <li>• Crear un comentario para el cocinero.</li> <li>• Crear un comentario sobre un plato.</li> <li>• Valorar un plato.</li> </ul>	
Referencias	UR_CC_04, UR_CC_08, UR_CR_01, UR_CR_03, UR_CR_04	

Tabla 95 – Componente Gestor de comentarios

VISUALIZADOR COMENTARIOS CLIENTE		COM_04
Propósito	Mostrar los comentarios en la interfaz de cliente del restaurante.	
Función	<ul style="list-style-type: none"> <li>• Mostrar en la carta comentarios hechos por otros clientes.</li> <li>• Mostrar los comentarios propios hechos sobre un plato tras añadirlo al pedido.</li> </ul>	
Referencias	UR_CC_02, UR_CC_09	

Tabla 96 – Componente Visualizador de comentarios de cliente.

GESTOR DE COMUNICACIÓN CLIENTE		COM_05
Propósito	Gestionar el envío y recepción de datos entre cliente del restaurante y servidor.	
Función	<ul style="list-style-type: none"> <li>• Enviar información al servidor sobre: <ul style="list-style-type: none"> <li>○ Platos pedidos.</li> <li>○ Comentarios para la cocina.</li> <li>○ Comentarios sobre platos.</li> </ul> </li> <li>• Pedir al servidor comentarios de los platos.</li> </ul>	
Referencias	UR_CC_01, UR_CC_02, UR_CC_07, UR_CC_08, UR_CC_09, UR_CC_11	

Tabla 97 – Componente Gestor de comunicación de cliente

GESTOR DE COMUNICACIÓN MANAGER		COM_06
Propósito	Gestionar el envío y recepción de datos entre manager y servidor.	
Función	<ul style="list-style-type: none"> <li>• Enviar información al servidor sobre: <ul style="list-style-type: none"> <li>○ Pedidos finalizados.</li> <li>○ Importe del pedido.</li> </ul> </li> <li>• Pedir al servidor: <ul style="list-style-type: none"> <li>○ Pedidos activos.</li> <li>○ Pedidos finalizados.</li> <li>○ Comentarios para la cocina.</li> <li>○ Importe del pedido.</li> </ul> </li> </ul>	
Referencias	UR_MC_01, UR_MC_02, UR_MC_04, UR_MC_05, UR_MC_06, UR_MC_07	

Tabla 98 – Componente Gestor de comunicación manager

VISUALIZADOR DE PEDIDOS MANAGER		COM_07
Propósito	Mostrar los pedidos realizados por clientes del restaurante.	
Función	<ul style="list-style-type: none"> <li>• Mostrar listado de pedidos activos.</li> <li>• Mostrar historial de pedidos terminados.</li> <li>• Mostrar pedidos que solicitan realizar el pago de la cuenta.</li> <li>• Mostrar información del pedido: <ul style="list-style-type: none"> <li>○ Identificador de pedido.</li> <li>○ Platos del pedido.</li> <li>○ Precio.</li> <li>○ Fecha.</li> <li>○ Hora.</li> <li>○ Número de comensales.</li> </ul> </li> </ul>	
Referencias	UR_MC_01, UR_MC_03, UR_MC_06	

Tabla 99 – Componente Visualizador de pedidos de manager

FILTRO DE BÚSQUEDA		COM_08
Propósito	Filtrar la búsqueda del historial de pedidos realizados.	
Función	<ul style="list-style-type: none"> <li>• Filtro de búsqueda por: <ul style="list-style-type: none"> <li>○ Número de mesa.</li> <li>○ Identificador de pedido.</li> <li>○ Plato incluido en el pedido.</li> <li>○ Precio del pedido.</li> <li>○ Fecha del pedido.</li> </ul> </li> </ul>	
Referencias	UR_MC_07	

Tabla 100 – Componente Filtro de búsqueda

GESTOR DE PEDIDOS ACTIVOS		COM_09
Propósito	Gestionar los pedidos activos y finalizados.	
Función	<ul style="list-style-type: none"> <li>• Terminar un pedido activo.</li> <li>• Marcar platos de un pedido activo.</li> <li>• Marcar un pedido activo como pagado.</li> <li>• Actualizar interfaz de pedido activo.</li> </ul>	
Referencias	UR_MC_02, UR_MC_04, UR_MC_05, UR_MR_01, UR_MR_02	

Tabla 101 – Componente Gestor de pedidos activos

La siguiente matriz de trazabilidad verifica que todos los requisitos han sido recogidos por los componentes, ya que si algún requisito quedara sin ser asignado a al menos un componente, significaría que el sistema no ha tenido en cuenta dicho requisito y habría de solucionarlo de inmediato.

	COM_01	COM_02	COM_03	COM_04	COM_05	COM_06	COM_07	COM_08	COM_09
UR_CC_01	X				X				
UR_CC_02		X		X	X				
UR_CC_03	X								
UR_CC_04			X						
UR_CC_05	X								
UR_CC_06	X								
UR_CC_07					X				
UR_CC_08			X		X				
UR_CC_09				X	X				
UR_CC_10		X							
UR_CC_11					X				
UR_CR_01	X	X	X						
UR_CR_02		X							
UR_CR_03		X	X						
UR_CR_04	X		X						
UR_MC_01						X	X		
UR_MC_02						X			X
UR_MC_03							X		
UR_MC_04						X			X
UR_MC_05						X			X
UR_MC_06						X	X		
UR_MC_07						X		X	
UR_MR_01									X
UR_MR_02									X

Tabla 102 – Matriz de trazabilidad componentes y UR.

### 3.2.2 Diseño de la Base de Datos

En el proyecto, es necesario almacenar cierto tipo de información para permitir que, tanto el cliente como el manager, tengan acceso a la misma. Además, se debe mantener un historial de toda esa información para ser consultada en cualquier momento. Para facilitar esta tarea, se utilizará MySQL como gestor de base de datos.

Al tratarse de una aplicación distribuida, cada vez que uno de los clientes necesite información de base de datos enviará la orden al servidor. Será el servidor quien haga la petición, en lenguaje SQL, a base de datos y devuelva la información al cliente que lo solicitó.

A continuación se detalla el modelo Entidad-Relación, también llamado E/R, que modela los datos y relaciones de base de datos. En el diagrama se reflejan las entidades del sistema, que están representadas con un rectángulo, y las relaciones existentes entre ellas, con un rombo. Por simplicidad en cuanto a lectura, se ha evitado la representación de todos los atributos, los cuales serán descritos más abajo.

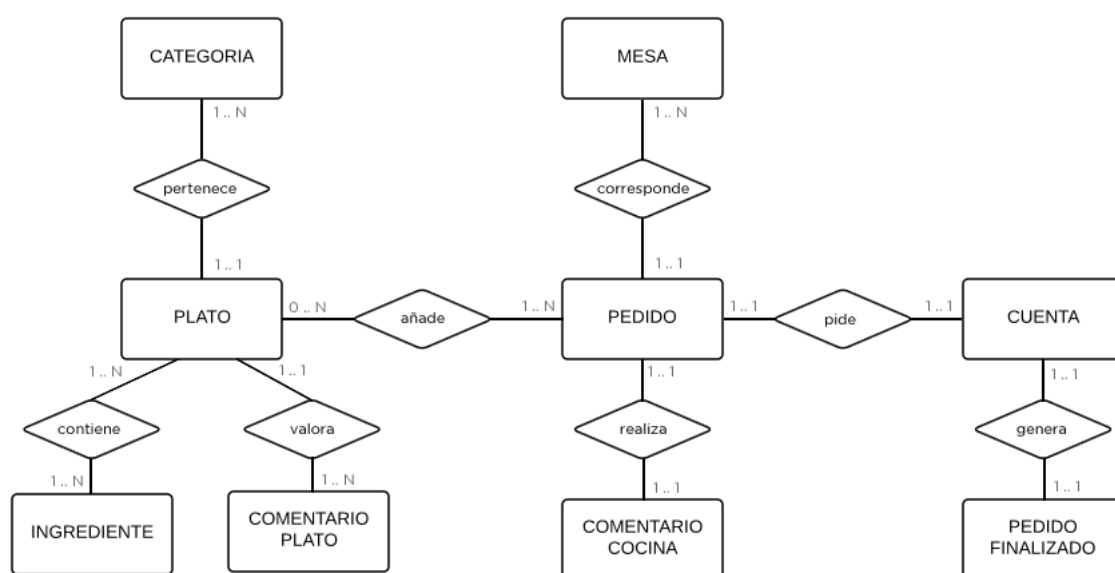


Figura 20 – Diagrama E/R.

Una vez visto el modelo ER y las relaciones existentes entre entidades, se explica en detalle la necesidad de existencia de cada tabla, así como de los atributos que la conforman y qué tipo de acceso reciben por parte de los usuarios.

- **MESA:** Esta tabla almacena la información del identificador de pedido y comensales asociados a cada mesa. Cada entrada contiene los siguientes campos:
  - **id\_mesa:** clave unívoca que identifica cada entrada en la tabla. Es asignado automáticamente por la base de datos.
  - **num\_mesa:** identificador del número de mesa en el restaurante. Este campo, en combinación con **id\_pedido**, se comporta como clave alternativa.
  - **id\_pedido:** identificador del pedido que se va a realizar durante la sesión activa. Es una clave externa, perteneciente a la tabla *Pedido*.
  - **comensales:** número de comensales para la sesión.

Los accesos que se realizan sobre esta tabla son de:

- **escritura:** se añade una entrada nueva cada vez que un cliente del restaurante introduce el número de comensales.
  - **lectura:** desde la aplicación manager, tanto para mostrar información en detalle de un pedido, como para realizar búsquedas por filtro.
- **CATEGORÍA:** contiene el listado de categorías de platos que existen en la carta del restaurante. Cada entrada contiene los siguientes campos:
  - **id\_categoria:** clave unívoca que identifica cada entrada en la tabla. Es asignado automáticamente por la base de datos.
  - **nombre:** nombre de la categoría que se mostrará en la carta. También actúa como clave unívoca.
  - **posición:** pestaña donde se ubicará la categoría en la carta.

Los accesos que se realizan sobre esta tabla son de:

- **lectura:** desde la aplicación cliente.

- **PLATO:** contiene el listado de platos que puede consultarse en la carta del restaurante. Cada entrada contiene los siguientes campos:
  - **id\_plato:** clave unívoca que identifica cada entrada en la tabla. Es asignado automáticamente por la base de datos.
  - **nombre:** nombre del plato que se mostrará en la carta. También actúa como clave unívoca.
  - **descripción:** descripción del plato.
  - **imagen:** imagen del plato.
  - **ingredientes:** referencia a los ingredientes que contiene el plato.
  - **categoría:** referencia a la categoría a la que pertenece el plato.
  - **valoración:** referencia a los comentarios y valoraciones hechos por los clientes.

Los accesos que se realizan sobre esta tabla son de:

- **lectura:** desde la aplicación cliente.
- **INGREDIENTE:** contiene el listado de ingredientes de cada plato. Cada entrada contiene los siguientes campos:
  - **id\_ingrediente:** clave unívoca que identifica cada entrada en la tabla. Es asignado automáticamente por la base de datos.
  - **nombre:** nombre del ingrediente. También actúa como clave unívoca.
  - **imagen\_activo:** imagen que se muestra en la lista de ingredientes de la interfaz *Eliminar Ingrediente* cuando no se quiere eliminar.
  - **imagen\_inactivo:** imagen que se muestra en la lista de ingredientes de la interfaz *Eliminar Ingrediente* cuando se quiere eliminar.

Los accesos que se realizan sobre esta tabla son de:

- **lectura:** desde la aplicación cliente.



- **INGREDIENTE\_PLATO:** Esta tabla es una referencia cruzada que contiene qué ingredientes corresponden a qué platos. Cada entrada contiene los siguientes campos:
  - **id\_ingredient:** referencia al id del ingrediente de la tabla *Ingrediente*. Indica que el ingrediente se encuentra en el plato *id\_plato*.
  - **id\_plato:** referencia al id del plato de la tabla *Plato*. Indica que el plato contiene el ingrediente indicado en *id\_ingredient*.
  - **obligatorio:** indica si el ingrediente puede ser eliminado del plato o no.

Los accesos que se realizan sobre esta tabla son de:

- **lectura:** desde la aplicación cliente.
- **PEDIDO:** Esta tabla almacena los detalles del pedido que un cliente envía a cocina. Se añade una entrada con cada plato enviado a la cocina por los clientes. Cada entrada contiene los siguientes campos:
  - **id\_pedido:** id unívoco de la fila en la tabla. El valor no es asignado por base de datos. En su lugar, el *id\_pedido* es la composición de los campos correspondientes al número de mesa y a la fecha completa con formato YYYYMMDDhhmm, concatenados con un guión.
  - **nombre\_plato:** nombre de un plato pedido por el cliente.
  - **cantidad:** número de unidades del plato indicado pedido por el cliente.
  - **fecha:** fecha del pedido.
  - **terminado:** indicador de si el pedido ha sido terminado o no.
  - **precio:** precio unitario del plato pedido.

Los accesos que se realizan sobre esta tabla son de:

- **escritura:** se añade una entrada nueva cada vez que un cliente del restaurante envía la orden de preparación de un plato a cocina.

- **modificación:** el cliente puede incrementar la cantidad de platos pedidos, para lo cual es necesario modificar el valor de *cantidad* correspondiente al plato seleccionado. El campo *finalizado* cambia su valor una vez el cliente termina la sesión.
  - **borrado:** el manager puede terminar la sesión y borrar de esta tabla toda la información perteneciente a un pedido activo.
  - **lectura:** desde la aplicación manager se accede a esta tabla para recuperar la información de los pedidos activos en el restaurante.
- **PEDIDO\_TERMINADO:** Una vez el manager finaliza un pedido, esta tabla almacena la información de dicho pedido. Cada entrada se corresponde con la de los platos existentes en el pedido recién finalizado de un cliente por el manager. Todos los campos son escritos a partir de la copia de información de los campos homónimos existentes en la tabla *Pedido*.

La finalidad de esta tabla es la de mantener un historial de pedidos terminados y a la vez, que la tabla de pedidos activos se mantenga limpia, tan solo con platos activos.

Cada entrada de esta tabla contiene los siguientes campos:

- **id\_pedido:** id unívoco de la fila en la tabla.
- **nombre\_plato:** nombre de un plato pedido por el cliente.
- **cantidad:** número de unidades del plato indicado pedido por el cliente.
- **fecha:** fecha del pedido.
- **precio:** precio unitario del plato pedido.

Los accesos que se realizan sobre esta tabla son de:

- **escritura:** se añade una entrada nueva por cada plato existente en el pedido finalizado por el manager.
- **lectura:** desde la aplicación manager se accede a esta tabla para recuperar la información del historial de pedidos.

- **COMENTARIO\_PLATO:** Esta tabla almacena los comentarios y valoraciones realizados por los clientes sobre los platos del menú. Se añade una entrada cada vez que un usuario hace un comentario y valora con una puntuación entre 1 a 5 estrellas un plato que ha pedido en la sesión actual. Cada entrada de esta tabla contiene los siguientes campos:
  - **id\_comentario\_plato:** id unívoco de la entrada en la tabla. El valor del campo es asignado automáticamente por la base de datos.
  - **plato:** referencia al plato sobre el cual se ha realizado el comentario.
  - **comentario:** comentario introducido por el cliente desde la aplicación.
  - **valoración:** valoración del plato introducida por el cliente. El valor de este campo estará entre 1 y 5.

Los accesos que se realizan sobre esta tabla son de:

- **escritura:** se añade una entrada nueva por cada valoración y comentario realizados por un cliente.
  - **lectura:** desde la aplicación cliente, se leen las entradas de esta tabla para mostrar los comentarios y valoraciones en la carta del restaurante.
- **COMENTARIO\_COCINA:** Esta tabla almacena los comentarios que un cliente realiza sobre un plato del pedido para que sean tomados en cuenta por el cocinero. Se añade una entrada por cada plato con comentario que exista en el pedido, una vez este ha sido enviado a cocina. Cada entrada de esta tabla contiene los siguientes campos:
  - **id\_comentario\_cocina:** id unívoco de la entrada en la tabla. El valor del campo es asignado automáticamente por la base de datos.
  - **id\_pedido:** referencia al id del pedido al que se asocia el comentario.
  - **plato:** referencia al plato del pedido sobre el que se ha realizado el comentario.
  - **comentario:** comentario introducido por el cliente para el cocinero.

Los accesos que se realizan sobre esta tabla son de:

- **escritura:** se añade una nueva entrada por cada plato que contenga un comentario con instrucciones para el cocinero.
- **lectura:** desde la aplicación del mánager se recupera la información de la tabla para ser consultada desde la interfaz.
- **CUENTA:** Se añade una entrada cada vez que un cliente pide hacer el pago del pedido. El manager accederá a esta tabla cuando su interfaz muestre que un cliente ha pedido hacer el pago, y una vez se le haya cobrado, se encargará de notificarlo al servidor, que hará que el campo *pagado* se modifique y marque *SI*, además de establecer el precio final en caso de haber modificado el pedido desde la aplicación manager.
  - **id\_cuenta:** id unívoco de la entrada en la tabla. El valor del campo es asignado automáticamente por la base de datos
  - **id\_pedido:** referencia al id del pedido al que se asocia el comentario.
  - **pagado:** campo en el que queda registrado si un usuario ha hecho el pago del pedido o no. Tomará el valor *NO* hasta que el manager marque como pagado el pedido, en cuyo caso cambiará a *SI*.
  - **precio\_final:** valor del precio final del pedido completo.

Los accesos que se realizan sobre esta tabla son de:

- **escritura:** se añade una nueva entrada cuando el cliente pulse el botón *Pagar*.
- **modificación:** tanto el valor del campo *Pagado*, como el de *Precio\_final* pueden ser modificados por el manager desde su interfaz.
- **lectura:** desde la aplicación del mánager se recupera la información de la tabla para ser consultada desde la interfaz.

### **3.2.3 Diseño de Interfaz**

Este apartado define y explica el proceso de diseño de las interfaces del proyecto. Cada interfaz ha sido diseñada teniendo en cuenta el usuario final de la aplicación y en todo momento se ha intentado mostrar de la manera más clara posible la información que un usuario puede necesitar, así como las opciones disponibles.

Para la parte de cliente del restaurante, se desea crear una interfaz intuitiva y muy fácil de usar, ya que el cliente seguramente no sea un usuario experto y además en primera instancia será reacio al cambio de un camarero real por un dispositivo tablet. Por ello, deberemos facilitar la tarea de hacer un pedido lo máximo posible y a la vez hacer que el uso de la aplicación sea interesante y atractivo.

Por otro lado, para la parte del manager, se debe tener toda la información a mano de un solo vistazo, por lo que la interfaz principal de pedidos que se encuentran activos en el comedor del restaurante debe contener toda la información en una única pantalla. Por ello, solo la información estrictamente necesaria es mostrada, lo cual facilitará la tarea de organización en cocina.

A continuación se explica en detalle el proceso y las decisiones de diseño de las interfaces para cada aplicación.

#### **3.2.3.1 Aplicación del Usuario Cliente**

El diseño de la aplicación del usuario cliente se centra en el propósito de que toda la información esté muy integrada en pantalla, pero a la vez la interfaz no quede recargada ni se haga pesada a la vista.

Se da por hecho que el usuario de esta aplicación no es experto, por lo que se ha de mostrar una interfaz sencilla, con un número de interfaces muy limitado y donde se sepa siempre de dónde se viene, cómo se puede volver y qué opciones existen, de forma que el proceso completo de realizar un pedido sea intuitivo.

A continuación se muestran las interfaces de la aplicación cliente, junto con una breve explicación de las decisiones de diseño de cada una.

## Bienvenida y Selección de Comensales

La primera interfaz que verá el cliente del restaurante es la pantalla de bienvenida.

Esta interfaz está pensada para que el usuario haga una toma de contacto con la aplicación. En ella se le da la bienvenida, se muestra una imagen y un mensaje que le invita a seleccionar el número de comensales que ocuparán la mesa durante su visita. Es una interfaz sencilla con una dificultad de uso mínima.



Figura 21 – Aplicación cliente. Bienvenida y selección de comensales

En la aplicación veremos una paleta de colores en continuo contraste para permitir una mejor visualización de los elementos.

Se utiliza negro como fondo de todas las interfaces, ya que es sobre fondos blancos o negros sobre los que se obtienen mejores contrastes en contraposición con cualquier color, pero el color blanco puede resultar muy deslumbrante sobre pantallas de dispositivos tablet dependiendo de la luz ambiental, causando molestia visual en los usuarios, por lo cual se evitará el blanco empleando negro en su lugar.

Para las letras y botones, se emplearán tonalidades de gama fría, como son el azul, el verde o el morado que relajan la vista (en contraposición a lo que los tonos cálidos harían), pero sin permitir colores apagados que no llamen la atención, consiguiendo así por el contraste con el fondo que el usuario se dé cuenta de la existencia de botones y texto rápidamente.

## Carta del Restaurante

Tras realizar la selección de comensales y pulsar el botón *Continuar*, se muestra la carta del restaurante en una nueva interfaz.

Por lo general, en una carta de restaurante encontramos varias hojas donde aparecen listados de platos que suelen estar agrupados por categorías (bebidas, entrantes, ensaladas, carnes, postres...), seguidos de una breve descripción y el precio, del modo que se muestra a continuación.



Figura 22 – Ejemplo de carta tradicional en papel de un restaurante.

En la aplicación cliente, el reto de esta interfaz era introducir en una única pantalla toda la información de la carta que el cliente vería en papel en un restaurante, sin que la cantidad de datos abrume al usuario y sienta rechazo al usar la aplicación.

Para conseguir el objetivo, se diseña una interfaz dividida en tres secciones, en las cuales se muestra toda la información como se describe a continuación.

- **Categorías:** Muestra las categorías de platos de la carta. Al seleccionar una de las categorías, se muestran todos los platos pertenecientes a la misma.
- **Listado de platos:** Muestra un listado de todos los platos que podemos añadir al pedido, dentro de la categoría seleccionada. Al seleccionar uno de los platos, en la sección de detalles del plato veremos la información del mismo.
- **Detalles del plato:** Ofrece información del plato que tenemos seleccionado y permite acceder a las opciones disponibles sobre dicho plato. Más adelante veremos en detalle el diseño de esta parte.

Además de estas tres secciones, en la parte inferior por lo general siempre estarán dispuestos los botones de acción, como en este caso son el botón de añadir y el botón ver pedido.

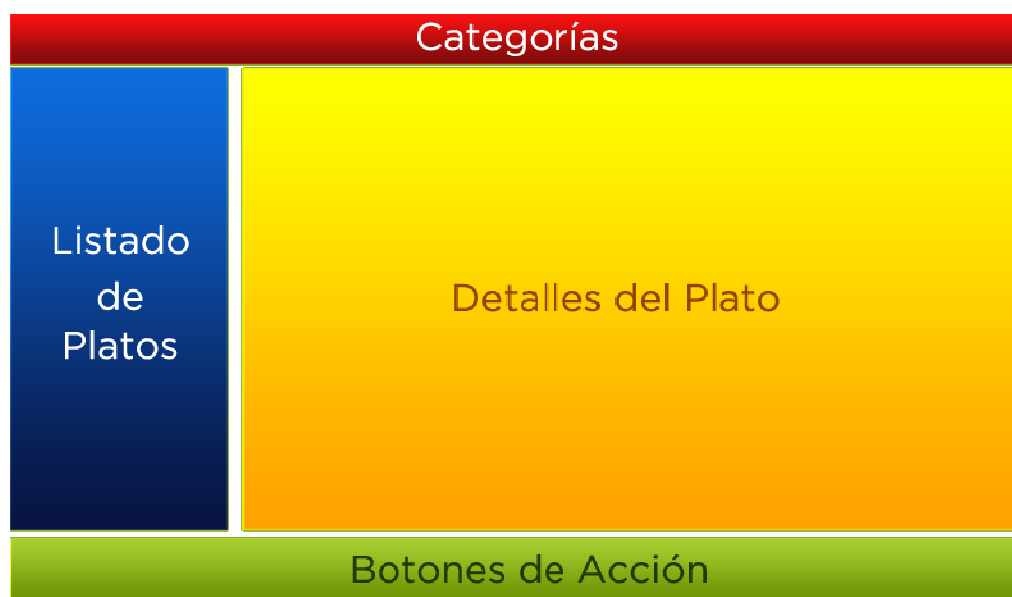


Figura 23 – Esquema de la interfaz de la carta.

En la parte de botones de acción, los botones se distribuyen a lo largo del eje horizontal, ubicándose dependiendo de la relación entre la interfaz y el botón. Por ejemplo, el botón *Añadir* se ubica en la sección de detalles del plato, ya que para el cliente es intuitivo mirar esta parte de la pantalla tras leer toda la información. El color verde y el símbolo positivo junto al texto *Añadir*, anima al cliente a pulsarlo.



Por otro lado, el botón *Ver pedido* se saca del contexto del plato y se ubica bajo el listado de pedidos, ya que nada tiene que ver con los detalles del plato y así se evita confusión entre los dos botones de acción de esta interfaz. Se mantiene el mismo color que para el botón *Añadir* para no recargar la interfaz con una paleta de colores extensa.

Esta es la vista real de la interfaz de la carta en la aplicación cliente, donde se reproduce el esquema mostrado anteriormente, pero con los componentes reales que el cliente vería al hacer uso de la aplicación.



Figura 24 – Aplicación cliente. Carta del restaurante.

La sección que ha requerido un cuidado minucioso a la hora de ser diseñada, es la de *Detalles del plato*. Ocupa la mayor parte del espacio, ya que se ubican todos los detalles de un plato y para evitar recargar la vista, se disponen bastantes espacios vacíos que separan la información.

De esta sección, el usuario siempre tendrá a la vista el nombre, la imagen y el precio del plato.

Además de ello, podrá acceder a la descripción del mismo, a las valoraciones y comentarios hechos por otros usuarios si los hubiera y a indicadores que muestran si el plato es apto para vegetarianos y/o para celíacos.

Por cuestiones de diseño, se quiso que esta información estuviera disponible al navegar entre los distintos platos y se pudiera ver de un vistazo, pero que al mismo tiempo se pudiera ocultar para dar paso a las opciones de *Eliminar ingrediente* e *Introducir comentario* para el cocinero.

De este modo la interfaz queda más limpia, con menos información en pantalla y diferenciando lo que es información del plato de lo que son las opciones que se pueden realizar.

Para iterar entre estas dos vistas, se diseña un sistema de pestañas verticales, que no interfieran ni se confundan con las pestañas correspondientes a las categorías, de manera que cada pestaña muestre o bien la información o las opciones del plato.

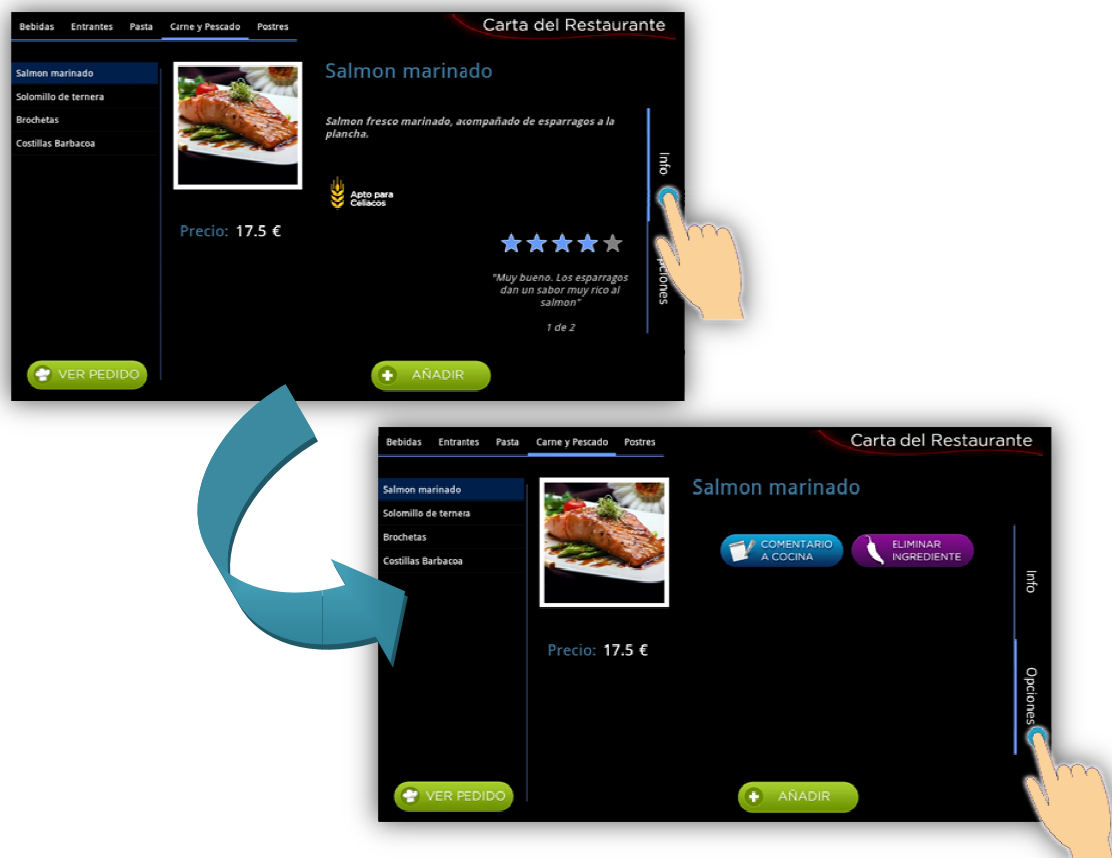


Figura 25 – Transición al iterar entre Info y Opciones de un plato.

Seleccionando la pestaña *Opciones*, la información del plato se oculta y se muestran dos botones que permiten realizar un comentario para el cocinero, o eliminar ingredientes del plato previo su pedido.

Si el botón *Comentario a Cocina* es pulsado, aparecerá un cuadro de texto bajo ambos botones y se desplegará un teclado en pantalla que el cliente deberá usar para introducir el comentario deseado. Se decide integrar el cuadro de texto en la misma pantalla por facilidad de uso para el cliente y para evitar mostrar más interfaces de las necesarias que le lleven a error o a no saber de dónde se procede.

Tanto antes como después de introducir un comentario, es posible eliminar ingredientes pulsando el botón *Eliminar Ingrediente*. Ello dará paso a una nueva pantalla en la que se muestran los ingredientes del plato que pueden ser eliminados.



Figura 26 – Aplicación cliente. Interfaz de eliminar ingredientes.

A pesar de querer limitar el número de pantallas lo máximo posible, se diseñó esta interfaz en una nueva pantalla para comodidad del cliente y por atractivo de la acción de eliminar ingredientes.

Se hubiera podido dibujar una lista con todos los ingredientes en la pantalla anterior y sobre ella marcar los no deseados, sin necesidad de crear una nueva

interfaz. Pero eso restaría vistosidad a la aplicación, ya que solo se verían listas por toda la interfaz.

En lugar de ello, se dispone esta interfaz en la que los ingredientes del plato quedan representados mediante un icono representativo del mismo y su nombre, quedando una interfaz llamativa.

Para eliminar el ingrediente, el cliente deberá tocar su icono, quedando dicho ingrediente marcado con un aspa roja y el fondo sombreado.



Figura 27 – Aplicación cliente. Eliminar ingrediente del plato.

Después de eliminar los ingredientes deseados, se dispone de un único botón en la parte inferior de la nueva interfaz que indica que ya están hechos todos los cambios. Por si misma la palabra *Hecho* podría no significar nada para el cliente, por lo que junto a ella se muestra una mano en posición de *OK* que sirve de aprobación y da a entender al cliente que ya no se quieren realizar más cambios.

Tras pulsarlo, la interfaz devuelve al usuario a la pantalla principal donde se encuentra la carta.

Una vez de vuelta en la carta, solo queda añadir el plato al pedido, que quedará anotado junto con los ingredientes eliminados y los comentarios introducidos.

## Vista de los Pedidos

Hasta este punto se han ido añadiendo distintos platos al pedido, pero el cliente además de añadir debe poder consultar y modificar el pedido realizado. Para ello, se diseña una interfaz a la que se accede desde la carta del restaurante a través del botón *Ver Pedido*.



Figura 28 – Selección del botón Ver Pedido desde la interfaz de la carta.

A la hora de ver el pedido realizado, ha de diferenciarse entre dos tipos de pedido.

- **Pedido en curso:** es decir, el que no se ha enviado aun a la cocina y sobre el cual el cliente puede realizar modificaciones
- **Pedido enviado:** aquel que ha sido confirmado para comenzar su preparación, el cual el cliente no puede modificar, pero si realizar ciertas operaciones como comentar los distintos platos o pagar y finalizar la sesión.

Teniendo esto en cuenta, se quiere que el usuario pueda consultar los datos del pedido diferenciando bien a qué tipo de pedido se hace referencia. Para ello, se diseña la interfaz de la vista de pedidos integrando dos pestañas en la parte superior de la misma, que permitirá navegar entre pedido en curso y pedido realizado.

Esto muestra al usuario que existe una diferenciación entre información del pedido y, al estar familiarizado desde la interfaz anterior con el sistema de pestañas, le resulta intuitiva la navegación entre ellas.

Las interfaces de cada pestaña tienen el mismo esquema, consistente en una única área de información que ocupa la totalidad de la pantalla, en la que se dispone un listado con los platos seleccionados, la cantidad y precio entre otros detalles.

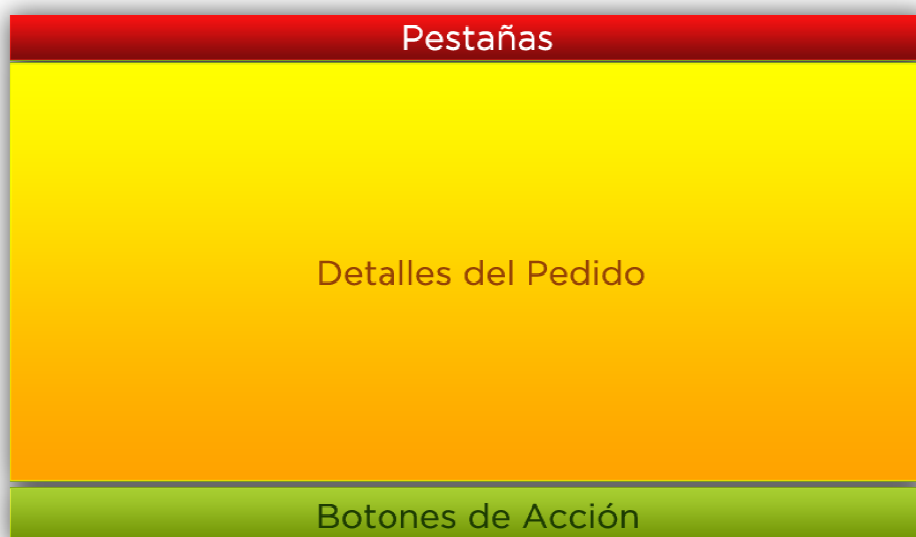


Figura 29 – Esquema de la interfaz de pedidos en la aplicación cliente.

Se diseña la interfaz para que sólo en el caso de haber enviado un pedido a la cocina, se muestra la pestaña correspondiente a la vista de pedidos realizados. En caso contrario, no se podrá navegar a la misma.

## ✓ Pedido en Curso

Por omisión, al seleccionar *Ver Pedido* desde la carta, se llega a la pantalla de pedidos en curso. Se diseña de este modo, ya que las dos acciones inmediatas a añadir platos son consultar el pedido antes de enviarlo y enviarlo una vez consultado, que son precisamente las acciones que se pueden realizar en esta pantalla.

No obstante y como se vio en el punto anterior, es importante permitir desde esta misma interfaz, acceder al pedido realizado, por lo que se dispone la pestaña correspondiente en la parte superior, siendo intuitiva la navegación hacia ella en caso de necesitarse.



Figura 30 – Aplicación Cliente. Vista de Pedido en curso.

Esta interfaz permite añadir o eliminar cantidades de platos pedidos. Para hacerlo intuitivo, se añadió un botón a la izquierda y derecha del número que indica la cantidad de platos actualmente en la lista en curso. Al pulsar cualquiera de los botones, el valor se modificara, al mismo tiempo que lo hace el precio.

Por otro lado, también es importante permitir al usuario consultar los comentarios para la cocina que se añadieron junto al plato.

Para no recargar la parte visible de la pantalla se decidió que, junto a los platos que tuvieran un comentario, se dispondría un icono indicativo que al presionarlo mostraría una ventana emergente en la pantalla con el comentario introducido.

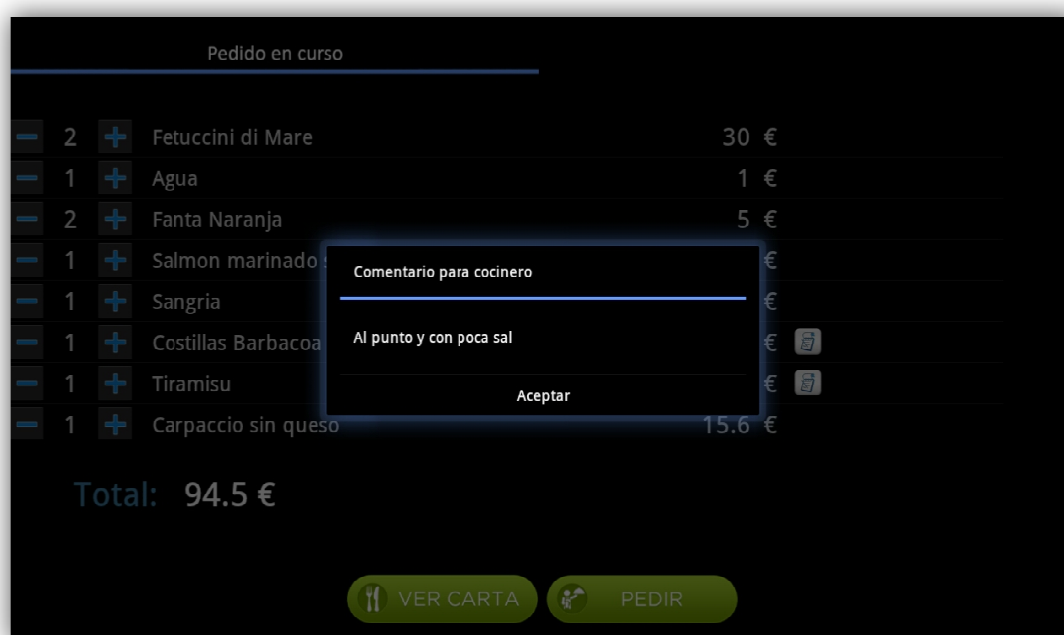


Figura 31 – Ventana emergente con el comentario introducido para el cocinero.

En la parte inferior de la interfaz se tienen dos botones. A la izquierda el botón para volver a ver la carta y a la derecha el botón que permite enviar el pedido a la cocina.

Ambos se han ubicado en el lado que por intuición representan, siendo la izquierda el lado que indica volver atrás, por tanto sugiriendo que la carta está una pantalla atrás, y la derecha un paso adelante, como en este caso sería enviar el pedido a la cocina.



Al seleccionar el botón pedir, aparece un mensaje de verificación en el centro de la pantalla para comenzar su preparación. Es importante este mensaje de confirmación, ya que de no existir se enviaría el plato al pulsar el botón *Pedir*, incluso si se pulsara por error.

Una vez se acepta el mensaje de confirmación de envío, entonces sí se realiza el envío del pedido a la cocina y se muestra un mensaje al cliente indicando que la cocina ha recibido la orden.

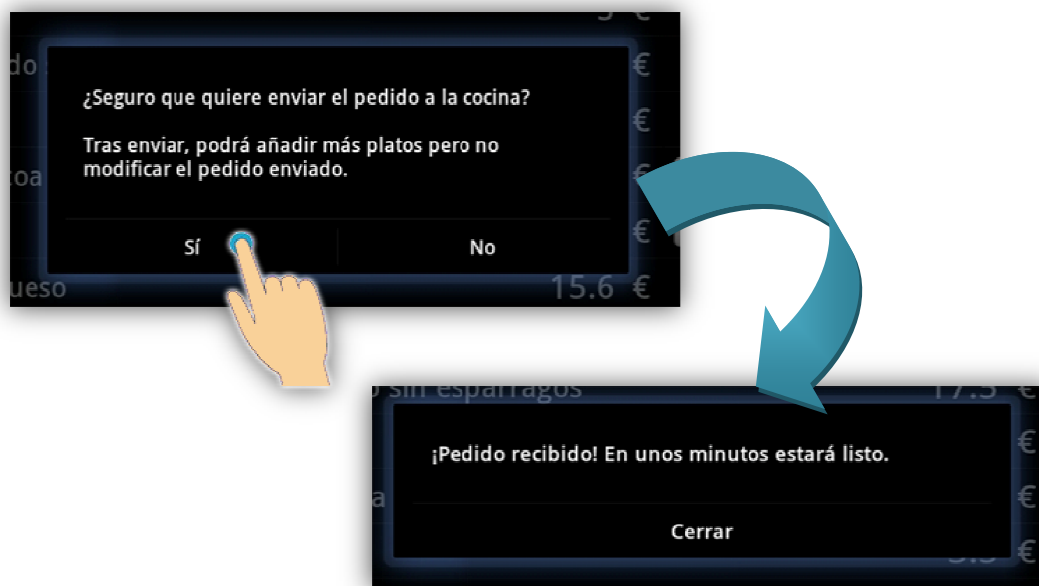


Figura 32 – Aplicación Cliente. Mensaje de confirmación al realizar un pedido.

### ✓ Pedido Realizado

La interfaz de pedido realizado mantiene la misma estructura que la de pedido en curso, pero ahora no se permite incrementar o reducir la cantidad de platos, ni se muestran los comentarios hechos a cocina, ya que de nada sirve esta información al cliente una vez el pedido ha sido realizado.

Por el contrario, se introduce una nueva opción junto a cada plato que permite hacer una valoración del mismo, la cual será visible para el resto de clientes que vean la carta una vez se haya enviado el comentario.

Como el principal objetivo es evitar tener un gran número de pantallas, esta interfaz se diseña para que al pulsar sobre el icono junto al plato, aparezcan las opciones de valoración junto al listado de platos, sin necesidad de introducir más interfaces.



Figura 33 – Aplicación Cliente. Vista de Pedido realizado.

Si se valora un plato, ha de introducirse un comentario y una valoración entre una y cinco estrellas. Una vez escrito y seleccionado el número de estrellas, se pulsará el botón *Aceptar* para guardar la valoración. Un mensaje avisará de que el comentario ha sido recibido, agradeciendo al cliente su colaboración.

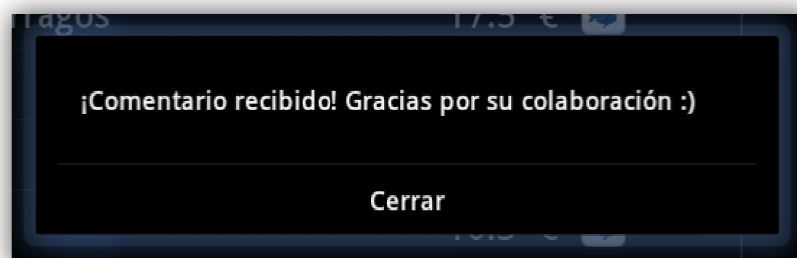


Figura 34 – Mensaje de confirmación a una valoración de plato recibida.

En la parte inferior, los botones de acción se ubican siguiendo la lógica de la interfaz del pedido en curso, estando *Ver Carta* a la izquierda y *Pagar* a la derecha.

Aunque a simple vista uno no se dé cuenta, el color azul transmite calma, tranquilidad y bienestar, y precisamente ha sido esa la razón por la cual se utiliza este color en el botón de pagar. Para entenderlo mejor, imaginemos el botón en rojo en lugar de azul. No sería la misma sensación, ¿verdad?

Al presionar el botón Pagar, se recuerda al cliente que la sesión terminará en caso de realmente querer pagar, dándole la opción de aceptar y terminar la sesión, o cancelar la acción y volver a la interfaz de pedidos realizados.

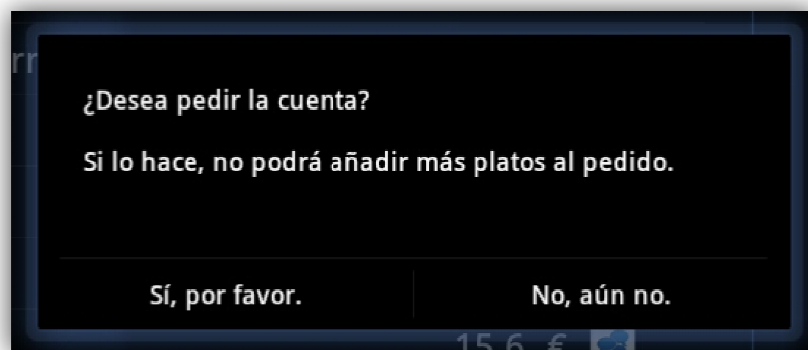


Figura 35 – Mensaje de confirmación para pedir la cuenta del pedido.

Para finalizar la sesión de un cliente, se muestra un mensaje de despedida y agradecimiento por su visita.

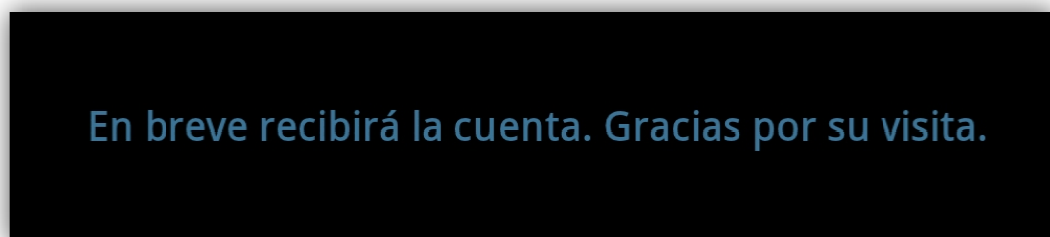


Figura 36 – Mensaje de despedida y agradecimiento.

### 3.2.3.2 Aplicación del Usuario Manager

La aplicación desarrollada para el manager del restaurante tiene dos objetivos: mostrar en una única interfaz toda la información de los pedidos activos en el restaurante y ofrecer la posibilidad de búsqueda de pedidos finalizados.

Aunque por lo general el manager mantendrá en pantalla la interfaz de pedidos en curso, se quiere facilitar la navegación al historial de pedidos para que pueda realizarla en cualquier momento. Por ello, siguiendo el estilo de pestañas visto para la aplicación cliente, la interfaz del manager contará con dos pestañas siempre visibles que permitirán la navegación entre Pedidos en curso e Historial de pedidos.

#### Inicio

Al comenzar la aplicación, el usuario será recibido con una pantalla de inicio con el logo de la aplicación, que se desvanecerá al tocar la pantalla, dando paso a la interfaz de pedidos en curso.

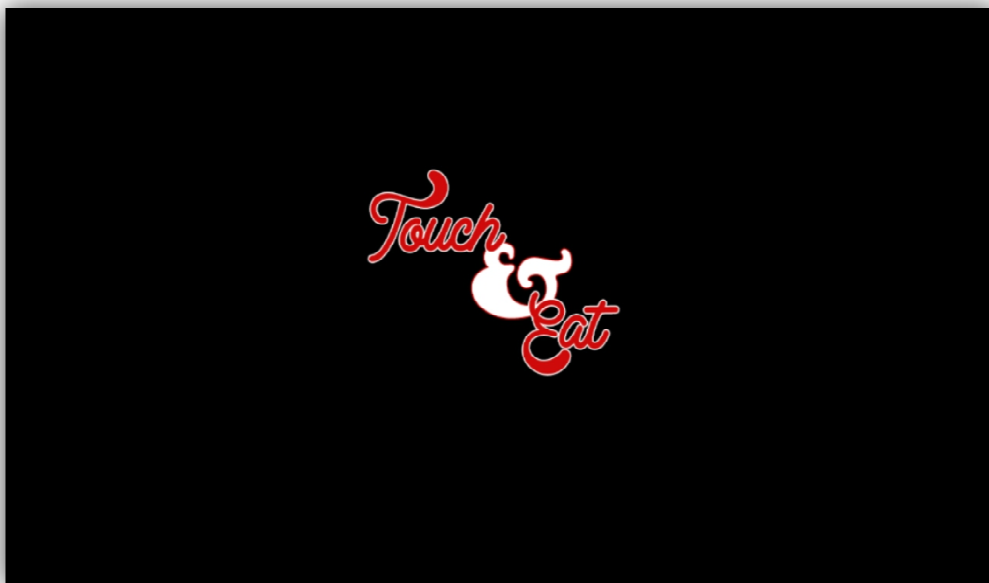


Figura 37 – Pantalla de Inicio.

## Pedidos en Curso

Para diseñar esta interfaz se tiene en cuenta que toda la información necesaria en cocina ha de aparecer en una única pantalla. Por esa razón, de nuevo se opta por dividir la pantalla en dos partes.

En la parte izquierda, se dispone un listado en el que se muestren los números de las mesas que han realizado un pedido y aún no ha sido finalizado.

Seleccionando cada mesa, en la parte derecha se muestran los detalles del pedido útiles para la gestión del mismo, como son el identificador de la mesa que ha pedido, el listado de platos con los ingredientes que han sido eliminados, los comentarios introducidos para el cocinero y el precio total al que asciende el pedido hasta el momento.

Para facilitar la gestión de los platos, se acompaña a cada uno de ellos de una casilla de verificación que el manager podrá marcar una vez el plato haya sido preparado y enviado a la mesa correspondiente.



Figura 38 – Aplicación Manager. Interfaz de Pedidos en Curso

Del mismo modo, cuando una mesa pida la cuenta desde su aplicación, se indica mediante un icono que aparecerá en el listado de la izquierda y al mismo tiempo se muestra un botón para marcar el pedido como pagado en la interfaz correspondiente a la mesa, que el manager deberá pulsar una vez la cuenta haya sido pagada.

Una vez seleccionado el botón de *Marcar pagado* y habiendo marcado todas las casillas de verificación del pedido, aparecer un botón que permite finalizar el pedido.



Figura 39 – Aplicación Manager. Vista de pedidos en curso al pedirse la cuenta.

Al finalizar el pedido, este se archiva y desaparece de la lista de pedidos activos.

Tanto en los pedidos en curso como en el historial de pedidos, siempre que se seleccione una mesa o un pedido, se tendrá a disposición un botón para imprimir la cuenta. Se decide así para cubrir cualquier caso en el que se necesite imprimir la factura, independientemente de si el cliente la ha solicitado o de si el pedido está finalizado o no.

## Historial de Pedidos

Esta sección permite ver todos los pedidos finalizados en el restaurante, con la opción de poder introducir filtros de búsqueda sobre el conjunto total de pedidos finalizados existente.

El diseño de esta interfaz tiene como propósito ofrecer una estructura muy similar a la de pedidos en curso, con la que el usuario ya está familiarizado, en la que además de mostrar pedidos, exista la posibilidad de introducir filtros de búsqueda.

Para conseguirlo, se diferencian dos secciones en la interfaz, correspondiendo la superior a los filtros de búsqueda y la inferior a los resultados obtenidos.

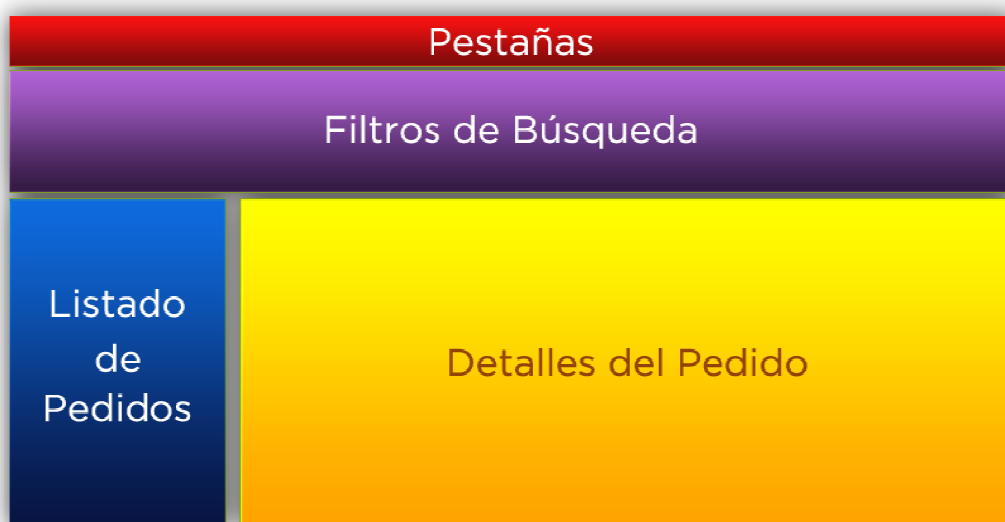


Figura 40 – Esquema de la interfaz Historial de búsqueda de la aplicación Manager

Como se ve más adelante, no es necesario introducir espacio adicional para los botones de acción, por lo que desaparece del esquema.

### ✓ Resultados del historial de pedidos

La parte correspondiente al resultado de una búsqueda se divide en un listado de pedidos y sus detalles correspondientes.

Entre estos detalles es importante destacar el número de mesa, los platos pedidos, la fecha y hora de realización del mismo, el precio total y si este fue pagado.

The screenshot displays the 'Historial de pedidos' (Order History) section of the Manager application. At the top, there are two tabs: 'Pedidos en curso' (Orders in progress) and 'Historial de pedidos' (Order history), with the latter being active. Below the tabs, there are search filters for 'Nº Mesa' (Table number), 'Plato' (Dish), 'Fecha Inicial' (Start date), 'Precio' (Price) with 'min' and 'max' sub-fields, 'Id Pedido' (Order ID), and 'Fecha Final' (End date). A 'BUSCAR' (Search) button is located on the right. The main content area shows details for 'Mesa 1' as of '2013/01/03 13:10'. A list of items is shown: 1 Agua, 2 Brocheta Vegetal sin berenjena, 3 Coca Cola Light, 1 Coca Cola Zero, 1 Fettuccini di Mare sin pimiento rojo sin tomate, and 2 Ravioli bolognesa sin jamon. At the bottom, there is an 'IMPRIMIR CUENTA' (Print bill) button, a 'PAGADO' (Paid) status indicator, and a 'Total: 166.60 €'.

Figura 41 – Aplicación Manager. Vista de historial de pedidos.

Esta interfaz cuenta con un único botón, correspondiente a la acción de *Imprimir Cuenta*. Tras introducir en la fase de diseño la sección destinada a filtros, el espacio en pantalla se ve reducido, por lo que se decide variar el diseño y por esta vez integrar este botón junto al precio final y el icono de *Pagado*, aprovechando mejor el espacio disponible.

Al presionar este botón en cualquiera de las dos interfaces, se muestra un mensaje que confirma que el ticket se está imprimiendo. A pesar de que por ahora no se dispone de una impresora compatible para comprobar la funcionalidad, se prevé que se complete el funcionamiento a corto plazo.

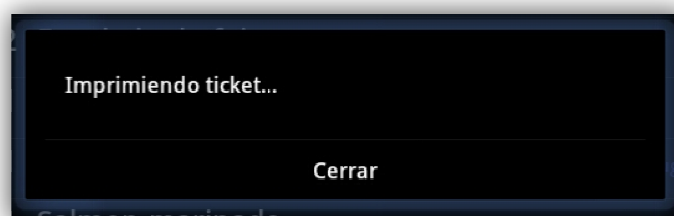


Figura 42 – Mensaje de confirmación al imprimir la cuenta.



## ✓ Filtros de búsqueda

Cada filtro disponible tiene un título descriptivo y junto a él un cuadro de texto que al seleccionarse muestra:

- Un teclado software para introducir el valor del filtro.
- O bien, un selector de fecha en caso de querer introducir un filtro por fecha inicial o final.

Podría haberse utilizado el teclado para todos los casos, pero el selector de fechas facilita en gran medida la tarea de selección al usuario, ya que es muy sencillo de usar y además muestra una vista de calendario del mes para una mejor ubicación.



Figura 43 – Aplicación Manager. Selección de fecha en filtro de búsqueda.

Para eliminar una fecha después de haberla, se diseña e implementa un botón en forma de aspa junto a cada cuadro de texto con formato fecha. De este modo, tras pulsarlo, la fecha correspondiente se eliminaría.

Una vez los filtros están establecidos, si se presiona el botón buscar se recuperan y muestran todos los pedidos con coincidencias en la parte inferior de la interfaz. En caso de no existir ningún pedido que cumpla con los filtros, se notificará al usuario con un mensaje y el espacio de resultados quedará vacío.

### 3.2.4 Diagrama de navegación

El diagrama de navegación consiste en la descripción gráfica del orden de aparición de pantallas que se producen durante la ejecución de la aplicación.

La aplicación cliente está orientada a usuarios de todo tipo, tanto expertos como inexpertos, por lo que la interfaz ha de ser descriptiva y la cantidad de pantallas limitada, de manera que siempre sepa de qué interfaz viene y a cual puede dirigirse.

Por ello, el diagrama de navegación es muy sencillo ya que, a pesar de dotar a la aplicación de toda la funcionalidad descrita en la sección de requisitos, es capaz de realizar cualquier operación desde cuatro interfaces principales, que corresponden con el siguiente diagrama.

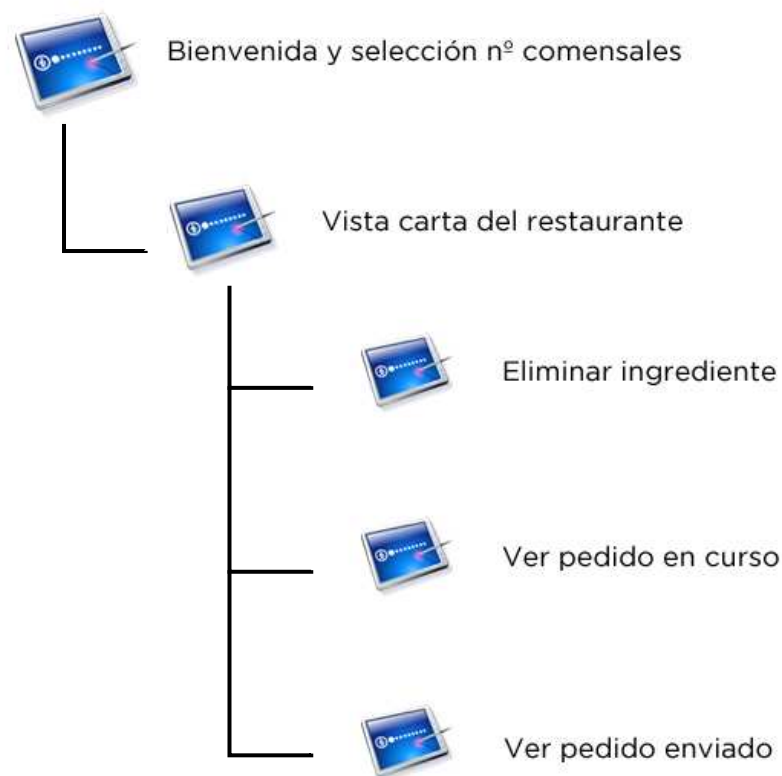


Figura 44 – Diagrama de navegación de la aplicación cliente.

Por otro lado, la aplicación manager está enfocada a ser utilizada por personal al que, a pesar de haber recibido un entrenamiento o curso rápido para aprender a utilizar la aplicación, deberá facilitársele en la medida de lo posible el acceso rápido y de un solo vistazo a toda la información que se esté moviendo en el restaurante.

Esa razón lleva a que todos los requisitos de usuario recogidos queden plasmados en dos únicas interfaces, accesibles cada una a través de un único movimiento. A continuación se muestra el diagrama de navegación de la aplicación manager.



Figura 45 – Diagrama de navegación de la aplicación manager.

Como vimos anteriormente en el capítulo dedicado al diseño de interfaz, ambos diagramas de navegación resumen el contenido de interfaces y navegación que se puede producir en el uso de ambas aplicaciones.

## 3.3 Implementación

La implementación del sistema consta de varios módulos diferenciados, entre los que se encuentran las aplicaciones cliente y manager, y el servidor que actúa en la comunicación entre ambas partes.

Todos los módulos han sido implementados desde cero y a continuación se detallan los aspectos más relevantes de este proceso.

### 3.3.1 Servidor

La comunicación entre aplicaciones se realiza a través de consultas al servidor, siendo este el encargado de recibir peticiones de las aplicaciones cliente y manager, y de responder con el envío del conjunto de datos correspondiente a dichas consultas.

La implementación del servidor cuenta con diferentes clases organizadas en paquetes cuyos detalles se muestran a continuación.

#### ✓ Clases Entidad

Para crear el servidor y que este sea capaz de comunicarse con las aplicaciones, se necesita conocer el tipo de objetos con los que tendrá que trabajar. Para ello, dentro de un paquete al que hemos denominado *.model*, se detallan las clases entidad para cada objeto del que se quiera recuperar o almacenar información.

Estas clases entidad contienen los atributos, constructores, *getters* y *setters* necesarios para definir un objeto.

#### ✓ Data Access Object (DAO)

Un DAO es un objeto que provee una interfaz abstracta a algunos tipos de base de datos u otros mecanismos de persistencia, permitiendo realizar distintas operaciones con información de base de datos, sin exponer los detalles de la misma.

En el servidor que estamos creando, deberá existir un DAO por cada clase entidad de las anteriormente creadas, ya que será el DAO el encargado de realizar

todas las operaciones necesarias sobre la base de datos. Por comodidad y organización, serán empaquetados en el paquete *.dao*.

La creación de estas clases incluye en su cuerpo diferentes métodos que se corresponden con las funciones básicas de persistencia en almacenamiento: las denominadas operaciones CRUD (Create, Read, Update y Delete).

### ○ Read

Para acceder a los datos contenidos en base de datos, en el objeto DAO se implementan métodos como *findall*, *findByName*, etc. Dependiendo de los datos que se quieran recuperar, se deberá crear uno u otro método.

Todos los DAO en el servidor disponen del método *findall*, que proporciona la lista completa de objetos del tipo solicitado. Si por ejemplo se hiciera un *findall* sobre *Dish*, se recuperaría cada una de las líneas de la tabla de platos existente en base de datos.

A continuación se muestra la estructura y se especifican los detalles más importantes de este método.

```
public List<Dish> findAll() {
    List<Dish> list = new ArrayList<Dish>();
    Connection c = null;
    String sql = "SELECT * FROM dish_order ORDER BY orderdate";
    try {
        Class.forName("com.mysql.jdbc.Driver");
        String url = "jdbc:mysql://localhost/touch_eat_db";
        c = DriverManager.getConnection(url, user, pass);
        Statement stmt = c.createStatement();
        ResultSet rs = stmt.executeQuery(sql);

        while (rs.next()) {
            list.add(processRow(rs));
        }
    } catch (
        ...
    ) finally {
        try{
            c.close();
        } catch (java.lang.Exception e){
            e.printStackTrace();
        }
    }
    return list;
}
```

Figura 46 – Código del método *findAll* correspondiente a DishDAO

Este es un extracto de la clase *DishDAO*, pero como se ha dicho, el método *findAll* se repite en todas las clases DAO.

La sentencia sql `"SELECT * FROM dish_order ORDER BY orderdate"` indica la operación de consulta que se realizará en la llamada a base de datos. En este caso, es una operación de selección del conjunto completo sobre la tabla *dish\_order* (la que contiene los platos pedidos y activos en el restaurante) ordenada por el campo fecha.

Dentro del bloque *try*, se hace la conexión con la base de datos *touch\_eat\_db*, introduciendo el usuario y contraseña necesario para realizar las conexiones. Es importante establecer ciertos parámetros de seguridad en las conexiones, como por ejemplo el usuario y contraseña, ya que como se vio anteriormente la seguridad del servicio REST tiene que ser reforzado, pues es muy débil.

```
Class.forName("com.mysql.jdbc.Driver");  
String url = "jdbc:mysql://localhost/touch_eat_db";  
c = DriverManager.getConnection(url, user, pass);
```

Figura 47 – Extracto de *findAll*. Código de conexión a base de datos.

Una vez establecida la conexión, se recupera la información de base de datos lanzando una petición mediante *executeQuery*, cuyo resultado queda guardado en un tipo de datos denominado *ResultSet*.

```
Statement stmt = c.createStatement();  
ResultSet rs = stmt.executeQuery(sql);
```

Figura 48 – Extracto de *findAll*. Código de ejecución de query.

Como lo que este método devuelve es una lista de platos, debemos recuperar la información guardada en el *ResultSet* anterior y almacenarla en una lista, que al final del método, una vez se haya cerrado la conexión, será devuelta.

Esta es la esencia de la operación de lectura sobre base de datos mediante un DAO. Para hacer consultas específicas sobre campos concretos de las tablas, se debe modificar la sentencia sql introduciendo los parámetros necesarios.

## ○ Create

Cuando se introducen nuevos elementos desde las aplicaciones a base de datos, se utiliza la sentencia *Create* del DAO correspondiente. En ella, se siguen los mismos pasos de conexión a base de datos que en el caso *Read*.

Antes de introducirse un elemento en base de datos, suelen hacerse ciertas comprobaciones, como que el dato no esté repetido, en cuyo caso en lugar de tratarse de una inserción, será una modificación. También puede ocurrir que al solicitarse la introducción de un elemento en una tabla, se requiera borrar dicho elemento de otra tabla. Esto ocurre al finalizar un pedido, en cuyo caso, todos los elementos del pedido desaparecen de la tabla de pedidos activos, y se introducen en pedidos terminados.

A continuación se muestra el código en el que se tratan todos los casos que pudieran ocurrir al hacer una inserción.

```
public Dish create(Dish dish) {
    Connection c = null;
    PreparedStatement ps = null;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        String url = "jdbc:mysql://localhost/touch_eat_db";
        c = DriverManager.getConnection(url, user, pass);
        if(findByIdName(dish.getIdorder(), dish.getnamedish()).isEmpty()) {
            ps = c.prepareStatement("INSERT INTO dish_order " +
                "(idorder, namedish, amount, orderdate, idtable, finished," +
                " nameingredientsdeleted, price) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)");
            ps.setString(1, dish.getIdorder());
            ps.setString(2, dish.getnamedish());
            ps.setString(3, dish.getamount());
            ...
            ps.executeUpdate();
        } else {
            if(dish.getfinished().compareToIgnoreCase("SI") == 0) {
                ps = c.prepareStatement("DELETE from dish_order WHERE idorder=? " +
                    "AND namedish=?");
                ps.setString(1, dish.getIdorder());
                ps.setString(2, dish.getnamedish());
            } else {
                ps = c.prepareStatement("UPDATE dish_order SET amount=?, finished=? " +
                    "WHERE idorder=? AND namedish=?");
                ps.setString(1, dish.getamount());
                ps.setString(2, dish.getfinished());
                ps.setString(3, dish.getIdorder());
                ps.setString(4, dish.getnamedish());
            }
            ps.executeUpdate();
        }
    }
}
```

Figura 49 – Extracto de código del método *create* correspondiente a *DishDao*

## ○ Update / Delete

Tal y como se ha visto en el caso anterior, *update* y *delete* siguen una estructura de código igual a *create*, donde lo único a modificar es la sentencia sql establecida para realizar estas nuevas operaciones.

Establecidas las operaciones que pueden realizarse mediante un DAO sobre la base de datos, solo queda ver a continuación de qué manera se hace la llamada a estos métodos.

## ✓ Acceso a recursos

En este momento todo está preparado para que, tras recibir una petición HTTP, se pueda hacer una llamada a las operaciones de creación, lectura, modificación y borrado vistas anteriormente.

Para ello se implementan nuevas clases utilizando Jersey, la implementación de JAX-RS, necesitando una clase por cada DAO existente. Serán ubicadas en el paquete *.resources*. La estructura de todas las clases es similar a la siguiente.

```
@Path("/dishes")

public class DishesResource {

    DishDao dishDao = new DishDao();
    DishFinishedDao dishFinishedDao = new DishFinishedDao();

    @GET
    @Produces({ MediaType.APPLICATION_JSON })
    public List<Dish> findAll() {
        return dishDao.findAll();
    }

    @POST
    @Produces(MediaType.APPLICATION_JSON)
    public Dish newDish(@FormParam("idorder") String id_order,
        @FormParam("namedish") String name_dish,
        @FormParam("amount") String amount,
        @FormParam("orderdate") String order_date,
        @FormParam("idtable") String id_table,
        @FormParam("finished") String finished,
        @FormParam("nameingredientsdeleted") String nameingredientsdeleted,
        @FormParam("price") String price,
        @Context HttpServletResponse servletResponse) throws IOException {

        Dish dish = new Dish(id_order, name_dish, amount, order_date,
            id_table, finished, nameingredientsdeleted, price);

        return dishDao.create(dish);
    }
}
```

Figura 50 – Código de la clase *DishesResource* con métodos GET y POST



Cada clase de acceso a recursos tendrá una anotación del tipo `@Path("/dishes")` al principio, cuya función es indicar la ubicación donde se encuentra el contenido al que se desea acceder. En este caso, la dirección apunta al directorio *dishes*, pero en cada clase deberá apuntar a su directorio correspondiente.

REST permite trabajar con peticiones HTTP como `@GET`, `@PUT`, `@POST`, `@DELETE` y `@HEAD`. En nuestro caso, utilizaremos `@GET` para las operaciones de recuperación, `@POST` para las operaciones de inserción y `@PUT` para las operaciones de modificación. No se necesita implementar una operación de borrado, ya que es el DAO el que se encarga de discernir entre la necesidad de hacer una modificación o un borrado al ser invocado.

### ○ @GET

Al recibir una petición HTTP de tipo GET, el servidor entiende que se le está pidiendo la recuperación de cierta información.

En el caso actual, cualquier petición GET que llegue sin parámetros, recupera todos los datos de la tabla correspondiente, y los devuelve en una lista. Si se indican parámetros, se llama al método correspondiente del DAO que permita una búsqueda selectiva sobre el conjunto de datos, devolviendo solo los que cumplan las condiciones especificadas.

En el siguiente extracto de código se ve cómo se haría una recuperación de todos los objetos de la tabla *bill* en el primer GET, mientras que en el segundo se recupera solo un objeto, que es el correspondiente al identificador de pedido introducido como parámetro.

```
@GET
@Produces({MediaType.APPLICATION_JSON})
public List<Bill> findAll() {
    return billDao.findAll();
}

@GET
@Path("/{idpedido}")
@Produces(MediaType.APPLICATION_JSON)
public Bill getBill(@PathParam("idpedido") String idPedido) {
    return billDao.findByBillPedido(idPedido);
}
```

Figura 51 – Tratamiento de peticiones HTTP de tipo GET

Analizando el código, bajo la anotación `@GET` aparecen otro tipo de anotaciones. En nuestro caso, serán de utilidad `@Path` y `@Produces` como vemos a continuación:

- **@PATH:** indica la ruta completa al directorio donde se encuentra el recurso que se quiere utilizar. En el caso de *idpedido*, se quiere acceder al elemento exacto cuyo identificador se corresponde con el pasado por parámetro.
- **@PRODUCES:** indica el tipo de datos que el método puede generar, generando un resultado en los tipos indicados automáticamente. Utilizaremos JSON únicamente como tipo ya que la información convertida a este tipo es muy fácil de manejar.

#### ○ @POST

Al recibir una petición HTTP de tipo POST, el servidor sabe que se le está pidiendo la inserción de información en base de datos.

Como se muestra en el siguiente fragmento de código, en el servidor las sentencias POST recibidas han de incluir todos los parámetros necesarios en la creación de un elemento de una tabla, ya que este método llama a *create* del DAO que crea e inserta un nuevo objeto asignando un valor a todos sus campos.

```
@POST
@Produces(MediaType.APPLICATION_JSON)
public Dish newDish(@FormParam("idorder") String id_order,
    @FormParam("namedish") String name_dish,
    @FormParam("amount") String amount,
    @FormParam("orderdate") String order_date,
    @FormParam("idtable") String id_table,
    @FormParam("finished") String finished,
    @FormParam("nameingredientsdeleted") String nameingredientsdeleted,
    @FormParam("price") String price,
    @Context HttpServletResponse servletResponse) throws IOException {

    Dish dish = new Dish(id_order, name_dish, amount, order_date,
        id_table, finished, nameingredientsdeleted, price);

    return dishDao.create(dish);
}
```

Figura 52 – Tratamiento de peticiones HTTP de tipo POST

Al igual que el caso anterior, también hace uso de ciertas anotaciones propias de Jersey, aunque en este caso `@Path` no se utiliza ya que las inserciones se harán siempre en la ubicación indicada al principio de la clase.

En cambio, se introduce el uso de la anotación `@FormParam` para indicar el nombre de cada campo en base de datos, de manera que se pueda relacionar y asignar el valor correspondiente en la creación del objeto.

### ○ @PUT

Del mismo modo que POST, la anotación PUT indica al servidor que se quiere realizar una inserción en base de datos.

La diferencia radica en que, en el caso del servidor actual, se utilizará PUT cuando se quiera realizar una operación de actualización o modificación, reservando la tarea de inserción completa solo a POST.

A continuación puede verse y compararse con el fragmento de código de POST, que la inserción de información es distinta aun realizándose sobre la misma tabla.

```
@PUT
@Produces(MediaType.APPLICATION_JSON)
public Dish updateDish(@FormParam("idorder") String id_order,
    @FormParam("finished") String finished,
    @Context HttpServletResponse servletResponse) throws IOException {

    Dish dish = new Dish(id_order, finished);

    return dishDao.update(dish);
}
```

Figura 53 – Tratamiento de peticiones HTTP de tipo PUT

De nuevo se utilizan anotaciones como `@Produces` y `@FormParam` del mismo modo que para POST, pero no es necesario introducir todos los parámetros como se hizo en el método de creación.

### 3.3.2 Aplicación cliente y manager

Se tratarán las dos aplicaciones desde el mismo punto de vista, ya que a pesar de tener objetivos distintos, la implementación es muy similar. En el desarrollo de ambas aplicaciones se destaca el módulo de comunicación, cuyo desarrollo se detalla a continuación.

#### ✓ Comunicación

En este momento, y disponiendo de un servidor capaz de recibir y procesar peticiones HTTP, es posible implementar el módulo de comunicación en las aplicaciones cliente y manager para realizar consultas y asegurar el correcto funcionamiento del sistema. Se organiza la funcionalidad en distintas clases.

##### ○ Clases Entidad

Las aplicaciones envían y reciben objetos JSON con la información de los pedidos. Para trabajar con esta información en Java, se utiliza una librería que permite asignar el valor del objeto JSON a una variable Java mediante anotaciones del tipo `@SerializedName("nombreserializado")`.

Además, se utiliza una clase adicional con la misma estructura que la anterior, que inserta cada elemento recuperado en una lista para mantener todos los resultados en un único conjunto con el que trabajar.

A continuación se muestra un ejemplo de ambas clases.

```
public class Mesa {  
  
    @SerializedName("nummesa")  
    public String num_mesa;  
  
    @SerializedName("idpedido")  
    public String id_pedido;  
  
    @SerializedName("numcomensales")  
    public String num_comensales;  
  
}  
  
public class MesaList {  
  
    @SerializedName("mesa")  
    public List<Mesa> mesa;  
  
}
```

Figura 54 – Entidad *Mesa* y entidad *MesaList*.

## ○ **Lógica REST**

El objetivo de la lógica REST es preparar la información antes de poder lanzar una llamada HTTP.

Se creará una única clase en la que se implementan todos los métodos que se necesitan para recuperar y enviar de información al servidor.

Existirán métodos de tipo get, para recuperar información, y de tipo set, para escribirla en base de datos.

### • **Métodos GET**

Estos métodos devuelven un objeto que se corresponde con la clase entidad explicada anteriormente, la cual permitía hacer una lista a partir de los resultados JSON obtenidos y convertidos.

Siendo así, se necesita un método que devuelva el tipo de lista correspondiente para la recuperación de información de distintas tablas.

La estructura del método es prácticamente igual en todas las llamadas, variando la URL de acceso y el tipo de dato a devolver, tal y como se muestra a continuación.

```
public static BillList getJasonBill() {
    String url = endpoint.concat("bill/");
    BillList resultadoOrden = null;

    InputStream source = RestManager.callGet(url);
    if(source != null) {
        Gson gson = new Gson();
        Reader reader = new InputStreamReader(source);

        resultadoOrden = gson.fromJson(reader, BillList.class);
    }

    return resultadoOrden;
}
```

Figura 55 – Método que prepara la llamada para obtener el listado de Cuentas.

La URL apunta a la ubicación raíz de la información, pero se necesita definir el espacio concreto (por verlo en modo base de datos, a la tabla concreta) al que se quiere acceder. Por ello, la primera línea de código en el método define la ruta completa de acceso al recurso que se recuperará (en este caso, hasta “*bill/*”).

Se aprecia una llamada a un método GET de la clase *RestManager* (que se explicará a continuación) al que se le transmite la URL. Como se verá, esta llamada devuelve un objeto *InputStream* que representa el objeto JSON con el resultado de la petición devuelto por el servidor.

Ese *InputStream* es tratado gracias a la librería GSON para ser convertido en un objeto Java de tipo lista cuyo contenido almacena los objetos de la clase entidad correspondiente (en el ejemplo superior, se tiene una lista de objetos de tipo *Bill*).

La lista de objetos Java obtenida como resultado es devuelta al invocador del método, que podrá usarlo para pintarlo en la interfaz o utilizarlo de cualquier otro modo.

- **Método SET**

Solo es necesario implementar un método SET para poder escribir cualquier objeto en base de datos.

Para este fin, se llama al método *CallSet* de *RestManager*, pasando como parámetro la dirección en la que se debe almacenar el recurso, y la lista con la información que se quiere guardar.

Como se ve a continuación, la estructura del código es muy sencilla.

```
public static void setJasonOrder(String directory, List<NameValuePair> lista) {  
    String url = inpoint+directory;  
    RestManager.callSet(url, lista);  
}
```

Figura 56 – Método que prepara la llamada para insertar

## ○ Gestor REST

Una vez toda la información ha sido tratada y está preparada, la clase gestora de REST envía una petición HTTP a la URL indicada por la lógica REST para que el servidor gestione una respuesta.

Esta clase contiene dos métodos que llevan a cabo todas las peticiones, siendo uno de recuperación de información y otro de escritura.

### • **callGet**

Como su nombre indica, el método hace una llamada GET de HTTP, para lo cual sólo necesita indicar la ruta donde se encuentra el o los recursos a recuperar.

Dicha ruta ha sido previamente establecida por la lógica REST, que es quien llama a este método *callGet* pasando la URL como parámetro.

Cuando el manager recupera la información, se la devuelve a la clase de lógica REST como un `InputStream`, de manera que pueda tratarla de forma adecuada como se explicó anteriormente.

A continuación se muestra el código que lleva a cabo esta tarea.

```
public static InputStream callGet(String url) {

    HttpClient httpClient = new DefaultHttpClient();
    HttpGet request = new HttpGet(url);

    try {
        HttpResponse result = httpClient.execute(request);
        final int statusCode = result.getStatusLine().getStatusCode();

        if (statusCode != HttpStatus.SC_OK) {
            System.out.println("Bad Response from server. Status "
                               + statusCode);
            return null;
        }

        HttpEntity responseEntity = result.getEntity();

        return responseEntity.getContent();
    }
}
```

Figura 57 – Código de recuperación de información en el Gestor REST.

- **callSet**

Este método hace una llamada SET de HTTP que permite la inserción de información en base de datos.

Para ello, además de la ruta donde se almacenará el recurso, se necesita enviar la lista con toda la información que se quiera insertar.

La lógica REST implementada se encarga de gestionar la URL y los recursos que se quieren introducir, por lo que este método solo tiene que recibirlo y generar la petición HTTP con la información provista para poder ejecutarla.

A continuación se muestra el código que lleva a cabo esta tarea.

```
public static void callSet(String url, List<NameValuePair> lista) {  
    HttpClient httpClient = new DefaultHttpClient();  
    HttpPost httpPost = new HttpPost(url);  
  
    try {  
        httpPost.setEntity(new UrlEncodedFormEntity(lista));  
        httpClient.execute(httpPost);  
    } catch (ClientProtocolException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Figura 58 – Código de inserción de información en el Gestor REST.



## 3.4 Pruebas

Una vez desarrollada la aplicación se ha de probar que funcione correctamente.

Para ello se desarrollan tests que aseguren el correcto funcionamiento del sistema comprobando que cada módulo funcione como es esperado, así como ciertos tests que permitan comprobar bajo qué condiciones el sistema tarda en presentar respuestas.

### 3.4.1 Pruebas de funcionalidad

Estas pruebas se desarrollan con el objetivo de verificar la funcionalidad de los distintos elementos que componen el sistema. Se realizan en un entorno controlado y comprobando que la respuesta de las operaciones coincidía con los requisitos establecidos en la fase de análisis.

Fueron repetidas a lo largo del desarrollo del proyecto y corrigiéndose los distintos errores encontrados cada vez, hasta obtener los resultados esperados.

Por longitud, solo se detallan algunas de las pruebas realizadas. Para facilitar la lectura, serán recogidas en tablas cuyo formato incluirá los siguientes campos:

- **Nombre:** nombre de la prueba con formato *Prueba XX*, donde XX será sustituido por el número correspondiente a la prueba.
- **ID:** identificador de la prueba. Su nomenclatura ha de seguir el formato PF\_XX, donde XX será sustituido por el número de la prueba.
- **Sistema:** aplicación sobre la cual se realiza la prueba.
- **Resultado:** especifica si tras la prueba se ha obtenido el resultado esperado. Puede tomar los valores de *Válido* e *Inválido*.
- **Descripción:** descripción del objetivo de la prueba.
- **Acciones:** pasos a seguir para la realización de la prueba.

Prueba Funcional 1			PF_01
Sistema	Cliente / Manager	Resultado	Válido
Descripción	Comprobación de conexión de cliente con el servidor y establecimiento de parámetros de inicio.		
Acciones	<ol style="list-style-type: none"> <li>1. Iniciar la aplicación cliente.</li> <li>2. Introducir dirección IP y puerto del servidor y continuar.</li> <li>3. Comprobar que no aparecen mensajes de error en la aplicación.</li> <li>4. En el caso de cliente, comprobar que el número de comensales se guarda en la dirección especificada.</li> </ol>		

Tabla 103 – Prueba de funcionalidad PF\_01

Prueba Funcional 2			PF_02
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que el cliente no puede salir de la aplicación con el botón de volver a atrás de la barra de estado del dispositivo.		
Acciones	<ol style="list-style-type: none"> <li>1. Iniciar y configurar la aplicación cliente hasta llegar a la carta.</li> <li>2. Pulsar el botón de volver a atrás de la barra de estado del dispositivo.</li> <li>3. Introducir cualquier cadena de caracteres en el cuadro de texto y pulsar aceptar.</li> <li>4. Comprobar que la aplicación no se cierra a no ser que se introduzca la contraseña de administrador.</li> </ol>		

Tabla 104 – Prueba de funcionalidad PF\_02

Prueba Funcional 3			PF_03
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que la interfaz muestre el listado de categorías, los platos correspondientes a cada categoría, los detalles correspondientes a cada plato y las operaciones a realizar.		
Acciones	<ol style="list-style-type: none"> <li>1. Iniciar y configurar la aplicación cliente hasta llegar a la carta.</li> <li>2. Pulsar las diferentes categorías y comprobar que el listado de platos se actualiza correctamente.</li> <li>3. Pulsar los platos del listado y comprobar que los detalles de cada plato se actualizan correctamente.</li> <li>4. Pulsar las pestañas de opciones y seleccionar cada botón.</li> </ol>		

Tabla 105 – Prueba de funcionalidad PF\_03

Prueba Funcional 4			PF_04
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que un comentario a cocina pueda añadirse antes de eliminar ingredientes.		
Acciones	<ol style="list-style-type: none"> <li>1. Iniciar y configurar la aplicación cliente hasta llegar a la carta.</li> <li>2. Acceder a las opciones del plato y pulsar <i>Comentario a cocina</i>.</li> <li>3. Introducir un comentario.</li> <li>4. Pulsar <i>Eliminar ingrediente</i>.</li> <li>5. Eliminar ingrediente y pulsar <i>Hecho</i>.</li> <li>6. Pulsar <i>Añadir plato</i>.</li> <li>7. Pulsar <i>Ver pedido</i>.</li> <li>8. Comprobar que el nombre del plato añadido contenga los ingredientes añadidos y que a su derecha figura el icono de comentario.</li> <li>9. Pulsar el icono de comentario y comprobar que en el mensaje emergente aparece el mensaje.</li> </ol>		

Tabla 106 – Prueba de funcionalidad PF\_04

Prueba Funcional 5			PF_05
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que no pueda pedirse un número negativo de platos.		
Acciones	<ol style="list-style-type: none"> <li>1. Añadir un plato al pedido.</li> <li>2. Ver el pedido.</li> <li>3. Pulsar el botón “-” tres veces.</li> <li>4. Comprobar que la cantidad nunca es inferior a cero.</li> </ol>		

Tabla 107 – Prueba de funcionalidad PF\_05

Prueba Funcional 6			PF_06
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que el listado de platos pedidos incluye todos los platos añadidos al pedido, incluyendo ingredientes eliminados.		
Acciones	<ol style="list-style-type: none"> <li>1. Añadir al menos nueve platos distintos al pedido.</li> <li>2. Desplazar el listado hacia arriba para comprobar que el último elemento no está repetido en la parte superior del listado.</li> </ol>		

Tabla 108 – Prueba de funcionalidad PF\_06

Prueba Funcional 7			PF_07
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que al realizar un pedido habiendo modificado las cantidades de un plato, estas cantidades sean correctas en el envío.		
Acciones	<ol style="list-style-type: none"> <li>1. Añadir un plato al pedido.</li> <li>2. Pulsar el botón <i>Ver pedido</i>.</li> <li>3. Pulsar al menos una vez el botón "+".</li> <li>4. Pulsar el botón <i>Enviar</i>.</li> <li>5. Comprobar en el pedido realizado la cantidad mostrada coincide con la establecida antes de realizar el envío.</li> </ol>		

Tabla 109 – Prueba de funcionalidad PF\_07

Prueba Funcional 8			PF_08
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que al enviar un pedido toda la información queda registrada.		
Acciones	<ol style="list-style-type: none"> <li>1. Añadir al menos un plato al pedido.</li> <li>2. Pulsar el botón <i>Ver pedido</i>.</li> <li>3. Pulsar el botón <i>Enviar</i>.</li> <li>4. Comprobar en base de datos que la información se ha recibido correctamente.</li> </ol>		

Tabla 110 – Prueba de funcionalidad PF\_08

Prueba Funcional 9			PF_09
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que una valoración de un plato quede registrada.		
Acciones	<ol style="list-style-type: none"> <li>1. Realizar un pedido y acceder a <i>Pedido realizado</i>.</li> <li>2. Pulsar el icono de escribir comentario de uno de los platos.</li> <li>3. Escribir un comentario.</li> <li>4. Pulsar un número de estrellas.</li> <li>5. Pulsar <i>Aceptar</i>.</li> <li>6. Comprobar en base de datos que la información se ha recibido correctamente.</li> </ol>		

Tabla 111 – Prueba de funcionalidad PF\_09

Prueba Funcional 10			PF_10
Sistema	Cliente / Manager	Resultado	Válido
Descripción	Comprobación de que los botones de acción y pestañas funcionen correctamente.		
Acciones	<ol style="list-style-type: none"> <li>1. Navegar por las interfaces probando que los botones de la parte inferior llevan a la interfaz correspondiente.</li> </ol>		

Tabla 112 – Prueba de funcionalidad PF\_10

Prueba Funcional 11			PF_11
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que no pueda realizarse la valoración de un plato si no se introduce texto y valoración en estrellas.		
Acciones	<ol style="list-style-type: none"> <li>1. Realizar un pedido y acceder a <i>Pedido realizado</i>.</li> <li>2. Pulsar el icono de escribir comentario de uno de los platos.</li> <li>3. Escribir un comentario.</li> <li>4. Pulsar <i>Aceptar</i>.</li> <li>5. Comprobar que no aparece el registro en base de datos.</li> <li>6. Eliminar el comentario y dejar espacios en blanco.</li> <li>7. Marcar al menos una estrella.</li> <li>8. Pulsar <i>Aceptar</i>.</li> <li>9. Comprobar que no aparece el registro en base de datos.</li> </ol>		

Tabla 113 – Prueba de funcionalidad PF\_11

Prueba Funcional 12			PF_12
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que al pedir la cuenta se disponga de la información en base de datos.		
Acciones	<ol style="list-style-type: none"> <li>1. Realizar un pedido y acceder a <i>Pedido realizado</i>.</li> <li>2. Pulsar el botón <i>Pagar</i>.</li> <li>3. Pulsar la opción afirmativa en el mensaje emergente.</li> <li>4. Comprobar en base de datos que la información de pago ha sido guardada.</li> </ol>		

Tabla 114 – Prueba de funcionalidad PF\_12

Prueba Funcional 13			PF_13
Sistema	Cliente	Resultado	Válido
Descripción	Comprobación de que al terminar una sesión, el identificador de la siguiente es distinto al de la previa.		
Acciones	<ol style="list-style-type: none"> <li>1. Iniciar la aplicación</li> <li>2. Consultar en base de datos el identificador de sesión.</li> <li>3. Realizar un pedido.</li> <li>4. Pagar el pedido.</li> <li>5. Iniciar otra sesión.</li> <li>6. Consultar en base de datos el identificador de sesión.</li> <li>7. Comprobar que ambos identificadores son distintos.</li> </ol>		

Tabla 115 – Prueba de funcionalidad PF\_13

Prueba Funcional 14			PF_14
Sistema	Manager	Resultado	Válido
Descripción	Comprobación de que los listados de platos muestran los platos correspondientes a la mesa o pedido indicado.		
Acciones	<ol style="list-style-type: none"> <li>1. Iniciar la aplicación manager.</li> <li>2. Pulsar pedidos en curso.</li> <li>3. Seleccionar una mesa que contenga una lista con al menos 9 elementos.</li> <li>4. Comprobar que el listado coincide con el de base de datos.</li> <li>5. Pulsar <i>Historial de Pedidos</i></li> <li>6. Seleccionar un pedido que contenga una lista con al menos 9 elementos.</li> <li>7. Comprobar que el listado coincide con el de base de datos.</li> </ol>		

Tabla 116 – Prueba de funcionalidad PF\_14

Prueba Funcional 15			PF_15
Sistema	Manager	Resultado	Válido
Descripción	Comprobación de notificación cuando un cliente pide la cuenta.		
Acciones	<ol style="list-style-type: none"> <li>1. Comprobar que en el listado de mesas aparece un icono junto a la mesa que ha pedido la cuenta.</li> <li>2. Comprobar que en los detalles de la mesa se ve el botón <i>Marcar Pagado</i></li> </ol>		

Tabla 117 – Prueba de funcionalidad PF\_15

Prueba Funcional 16			PF_16
Sistema	Manager	Resultado	Válido
Descripción	Comprobación de filtros.		
Acciones	<ol style="list-style-type: none"> <li>1. Acceder al <i>Historial de pedidos</i>.</li> <li>2. Introducir distintos filtros individualmente y en conjunto con otros filtros.</li> <li>3. Comprobar que los resultados obtenidos coinciden con los criterios de búsqueda.</li> </ol>		

Tabla 118 – Prueba de funcionalidad PF\_16

Prueba Funcional 17			PF_17
Sistema	Manager	Resultado	Válido
Descripción	Comprobación de marcado de casillas de verificación.		
Acciones	<ol style="list-style-type: none"> <li>1. Marcar varias casillas de verificación en <i>Pedidos activos</i>.</li> <li>2. Acceder al <i>Historial de pedidos</i>.</li> <li>3. Volver a <i>Pedidos activos</i>.</li> <li>4. Comprobar que las casillas siguen marcadas.</li> </ol>		

Tabla 119 – Prueba de funcionalidad PF\_17

### 3.4.2 Pruebas de estrés

Estas pruebas tienen como objetivo comprobar el funcionamiento del sistema bajo una gran carga de peticiones o limitaciones de la conexión. Con ello se establecen límites que no deben sobrepasarse para el correcto funcionamiento del sistema.

Para facilitar la lectura, serán recogidas en tablas cuyo formato incluirá los siguientes campos:

- **Nombre:** nombre de la prueba con formato *Prueba XX*, donde XX será sustituido por el número correspondiente a la prueba.
- **ID:** identificador de la prueba. Su nomenclatura ha de seguir el formato PS\_XX, donde XX será sustituido por el número de la prueba.
- **Descripción:** especifica los pasos llevados a cabo en la prueba.
- **Escenario:** posible caso real en el que puede ocurrir.
- **Resultado:** resume los resultados obtenidos de la realización de la prueba.

Prueba Estrés 1		PS_01
Descripción	Inserción en base de datos de 10, 20, 30 y 40 platos simultáneamente con una conexión Wi-Fi de 54Mbps.	
Escenario	Envío de un pedido con 10 platos en el mismo segundo de tiempo desde 1, 2, 3 y 4 mesas distintas, con una conexión óptima.	
Resultado	En cuanto a tiempos, se obtienen buenos resultados. <ul style="list-style-type: none"><li>• 10 platos: 0,6 segundos.</li><li>• 20 platos: 1,1 segundo.</li><li>• 30 platos: 2 segundos.</li><li>• 40 platos: 3,2 segundos.</li></ul>	

Tabla 120 – Prueba de funcionalidad PS\_01

Prueba Estrés 2		PS_02
Descripción	Inserción en base de datos de 10, 20, 30 y 40 platos simultáneamente con una conexión Wi-Fi de menos de 20Mbps.	
Escenario	Envío de un pedido con 10 platos en el mismo segundo de tiempo desde 1, 2, 3 y 4 mesas distintas, con una conexión de 20Mbps.	
Resultado	<p>En cuanto a tiempos, se obtienen buenos resultados.</p> <ul style="list-style-type: none"> <li>• 10 platos: 1,8 segundos.</li> <li>• 20 platos: 2,6 segundos.</li> <li>• 30 platos: 3,5 segundos.</li> <li>• 40 platos: 4,2 segundos.</li> </ul>	

Tabla 121 – Prueba de funcionalidad PS\_02

Prueba Estrés 3		PS_03
Descripción	Recuperación de 40, 80, 120 y 160 platos simultáneamente con una conexión Wi-Fi de 54Mbps.	
Escenario	Recuperación desde la interfaz del manager de 4 pedidos con 40 platos cada uno.	
Resultado	<p>En cuanto a tiempos, se obtienen muy buenos resultados.</p> <ul style="list-style-type: none"> <li>• 40 platos: 0,3 segundos.</li> <li>• 80 platos: 0,5 segundos.</li> <li>• 120 platos: 0,9 segundos.</li> <li>• 160 platos: 1 segundo.</li> </ul>	

Tabla 122 – Prueba de funcionalidad PS\_03

Prueba Estrés 4		PS_04
Descripción	Recuperación de 40, 80, 120 y 160 platos simultáneamente con una conexión Wi-Fi de menos de 10Mbps.	
Escenario	Recuperación desde la interfaz del manager de 4 pedidos con 40 platos cada uno.	
Resultado	<p>En cuanto a tiempos, se obtienen resultados no deseados.</p> <ul style="list-style-type: none"> <li>• 40 platos: 3,6 segundos.</li> <li>• 80 platos: 6,1 segundos.</li> <li>• 120 platos: 10,8 segundos.</li> <li>• 160 platos: 15,2 segundos.</li> </ul>	

Tabla 123 – Prueba de funcionalidad PS\_04

Con estas pruebas de estrés se demuestra que la limitación de velocidad en el caso del cliente no es tan acusada como en el caso del manager, que además realiza conexión y recuperación de datos de forma constante.



### 3.4.3 Matriz de trazabilidad de funcionalidad

La matriz de trazabilidad de funcionalidad tiene el objetivo de comprobar que los requisitos software recogidos en la sección de análisis hayan sido implementados en el sistema y funcionen correctamente.

Para ello, se marcan las intersecciones entre requisitos y pruebas en aquellas pruebas que permiten verificar que el requisito se ha cumplido. No es necesario que el objetivo final de la prueba coincida con el requisito señalado, es suficiente que durante la prueba se demuestre que el requisito se ha cumplido.

Siempre y cuando todas las filas de la matriz tengan al menos una marca, se podrá asegurar que las pruebas han validado la recogida del requisito en el sistema. En caso contrario, debería hacerse al menos una prueba específica para cada requisito no comprobado. Por extensión no se han incluido todas las pruebas realizadas, pero sí las más características y que en conjunto no dejen ningún requisito sin verificar.

	PF_01	PF_02	PF_03	PF_04	PF_05	PF_06	PF_07	PF_08	PF_09	PF_10	PF_11	PF_12	PF_13	PF_14	PF_15	PF_16	PF_17
SR_NFC_01	X												X				
SR_NFC_02			X														
SR_NFC_03			X	X													
SR_NFC_04			X														
SR_NFC_05			X														
SR_NFC_06			X														
SR_NFC_07			X														
SR_NFC_08			X	X						X							
SR_NFC_09			X	X						X							
SR_NFC_10			X	X						X							
SR_NFC_11										X							
SR_NFC_12										X							
SR_NFC_13										X							
SR_NFC_14					X	X	X			X							
SR_NFC_15					X		X										
SR_NFC_16					X					X							
SR_NFC_17							X	X		X							
SR_NFC_18										X							
SR_NFC_19								X	X		X						
SR_NFC_20										X	X						

[illegible]

Se puede ver que todas las filas han sido marcadas en al menos una ocasión, por lo que la funcionalidad recogida en los requisitos queda comprobada con el resumen de tests expuestos anteriormente.

# Capítulo 4

## Planificación y Presupuesto

---

En este capítulo se detalla la planificación que se seguirá a lo largo de las distintas fases del proyecto, así como el presupuesto del mismo. La planificación se acompaña de un diagrama de Gantt a modo de resumen gráfico del tiempo empleado en cada tarea. El presupuesto incluirá un desglose de todos los costes por separado para tener una mejor apreciación de los costes finales.

## 4.1 Planificación

En esta sección se trata el proceso de planificación seguido para la realización del proyecto.

Tanto las fiestas nacionales y provinciales como el periodo vacacional de 22 días repartido entre la última quincena de julio y la primera quincena de agosto se excluyen de la planificación.

Se muestra el resumen de actividades y el tiempo asociado a la realización de cada una de ellas, así como la fecha inicial y final en la que tuvo lugar cada actividad.

Actividad	Fecha inicio	Fecha fin	Duración (días)
Propuesta	12/12/11	12/12/11	1
Estudio Viabilidad	13/12/11	26/12/11	10
Documentación inicial	27/12/11	20/01/12	15
Análisis	23/01/12	08/02/12	13
Diseño	09/02/12	12/03/12	23
Implementación	13/03/12	01/10/12	122
Pruebas	02/10/12	15/10/12	10
Documentación	16/10/12	18/12/12	45

Tabla 125 – Planificación de actividades

### 4.1.1 Diagrama de Gantt

La planificación de actividades vista en el apartado anterior, se puede representar mediante un diagrama de Gantt como el que se muestra a continuación.

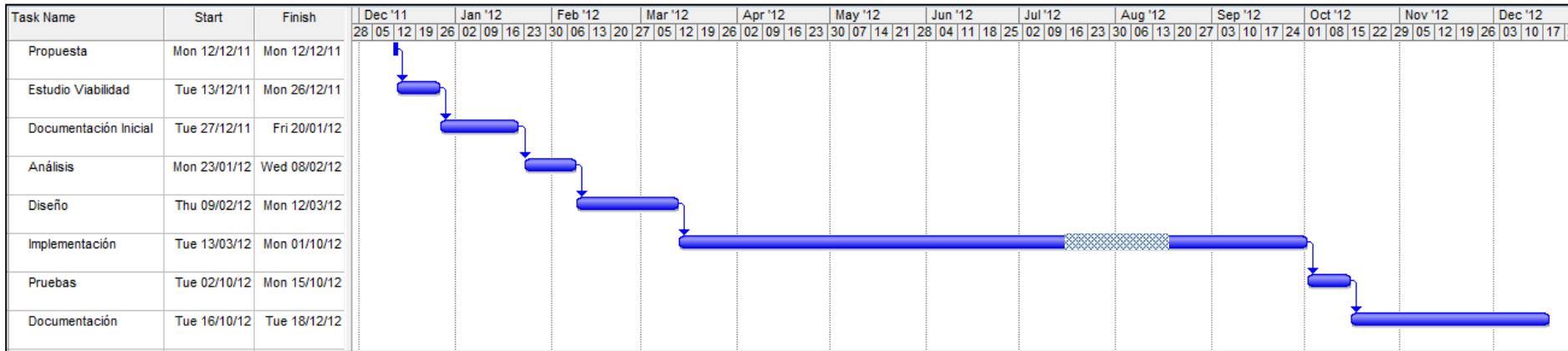


Figura 59 – Diagrama de Gantt para la planificación del proyecto

## 4.2 Presupuesto

En esta sección se detalla el presupuesto estimado para la realización del proyecto, desglosando los costes en función de su naturaleza.

Todos los costes se calculan sin I.V.A., este se aplica en el resumen de costes totales del proyecto que se muestra tras el cálculo detallado de cada parte.

### 4.2.1 Resumen de tiempo dedicado

El diagrama de Gantt desarrollado en el apartado anterior posibilita el cálculo del número de horas totales dedicadas al proyecto.

A continuación se muestra en detalle el cálculo de horas dedicadas por separado a cada tarea, y la suma en conjunto a la realización de todas ellas.

**Estudio de viabilidad:** 10 días x 8 horas = 80 horas.

**Documentación inicial:** 15 días x 8 horas = 120 horas.

**Fase de Análisis:** 13 días x 8 horas = 104 horas.

**Fase de Diseño:** 23 días x 8 horas = 184 horas.

**Fase de Implementación:** 122 días x 8 horas = 976 horas.

**Fase de Pruebas:** 10 días x 8 horas = 80 horas.

**Documentación final:** 45 días x 8 horas = 360 horas.

Tras realizar los cálculos, se comprueba que el tiempo total dedicado al proyecto es de 238 días, que se traducen en 1904 horas.

## 4.2.2 Desglose: Coste de personal

Se ha de calcular el coste de personal dependiendo del rol desempeñado y el tiempo invertido por cada uno de ellos en el desarrollo del proyecto [54] [55] . A continuación se muestra el desglose para cada tipo de rol.

Apellidos y Nombre	Categoría	Dedicación (hombre/mes)*	Coste hombre/mes (€)	Coste (€)
Fuentes Alba, Carmen	Analista	1	3.000	10.150
Fuentes Alba, Carmen	Diseñador	1	2.100	2.100
Fuentes Alba, Carmen	Programador	1	2.000	10.154
Total:				22.404

Tabla 126 – Desglose del coste de personal

\*) 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)  
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

Donde el coste ha sido calculado a partir de la siguiente ecuación.

$$\text{Coste personal} = \frac{\text{Total días} * \text{horas/día}}{\text{Dedicación hombre/mes}} * \text{Coste hombre/mes}$$

Siendo el resultado igual a 22.404 €, que es el coste total del personal dedicado al proyecto durante la duración completa del mismo.

## 4.2.3 Desglose: Coste de Hardware

Para el desarrollo del proyecto, ha sido necesaria la adquisición de un terminal portátil que servirá para el desarrollo de cada una de las fases del proyecto, además de la instalación del servidor, y de un dispositivo tablet para la realización de todas las pruebas de funcionamiento.

A continuación se detallan los datos de ambos dispositivos, para el cual se ha calculado el coste imputable a partir de la fórmula de amortización siguiente:

$$\frac{A}{B} * C * D$$

**A:** nº de meses desde la fecha de facturación en que el equipo es utilizado.

**B:** periodo de depreciación (60 meses).

**C:** coste del equipo (sin I.V.A).

**D:** % del uso que se dedica al proyecto.

Descripción	Coste (€)	% Uso dedicado al proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable (€)
Portátil Dell XPS 15'' Dual Core 2.4 GHz 4GB RAM 500GB Disco duro.	980	100	12	60	196
Samsung Galaxy Tab 10.1	489	100	12	60	97,8
<b>Total:</b>					<b>293,8</b>

Tabla 127 – Desglose del coste de Hardware

Como se resume en el total de la tabla anterior, el coste imputable correspondiente al hardware adquirido asciende a la cantidad de 293,8 €.

#### 4.2.4 Desglose: Coste de Software

A pesar de que la mayor parte del software con el que se ha trabajado es de código libre, y por tanto no tienen coste alguno, ha sido necesaria la adquisición de diversas licencias software para el desarrollo de ciertas fases del proyecto.

A continuación se detallan las licencias y el coste sin I.V.A. asociado.

Descripción	Coste (€)
Microsoft Office Professional 2007	499
Windows XP Professional	180
Adobe Photoshop CS5	250
<b>Total:</b>	<b>929</b>

Tabla 128 – Desglose del coste de Software

Como se resume en la tabla anterior, el coste total de software adquirido es de 929 €.



## 4.2.5 Resumen de costes

Tras analizar los distintos costes asociados al proyecto, se resume a continuación cada uno de ellos y se establece el coste total del proyecto. A la suma de costes calculada, se añade un 20% de costes indirectos para cubrir los riesgos del proyecto y los gastos que no han sido tenidos en cuenta al realizar el presupuesto.

Se detalla el coste total con y sin I.V.A.

Descripción de Costes	Coste Total (€)
Personal	22.404
Amortización	293,8
Costes de funcionamiento	929
Costes indirectos (20%)	4.725,36
Total sin I.V.A	28.352,16
<b>Total con I.V.A (21%)</b>	<b>34.306,11</b>

El coste total del proyecto sin I.V.A asciende a la cantidad de 28.352,16 € y aplicando el 21% de I.V.A correspondiente, el precio final del proyecto es de 34.306,11 €.



# Capítulo 5

## Conclusiones y Líneas Futuras

---

En este capítulo se detallan las conclusiones obtenidas tras la realización del proyecto, así como las tareas que se pretenden desarrollar en un futuro cercano para ampliar y mejorar la funcionalidad de la aplicación.

## 5.1 Conclusiones

Después de la realización del proyecto, es necesario destinar una parte del documento a analizar cuidadosamente el camino que ha llevado hasta aquí y ver qué objetivos han sido superados y que metas propuestas se han alcanzado.

Se hablará de ello tanto a nivel de proyecto como a nivel personal, poniendo por ahora un punto y aparte al desarrollo del prototipo.

### 5.1.1 Conclusiones del Proyecto

El objetivo del proyecto, consistente en la creación de un entorno que permitiera a un usuario hacer de forma autónoma un pedido en un restaurante a través de un dispositivo móvil y que este fuera recibido y accedido en tiempo real por el responsable de gestión de pedidos del restaurante, se ha conseguido satisfactoriamente.

El principio fundamental que se ha querido realizar para la presentación del prototipo ha sido, por un lado el establecimiento de comunicación Cliente-Servidor para el intercambio de información entre las dos aplicaciones, y por otro, mostrar una interfaz amigable, intuitiva y atractiva para ambos clientes que permita resolver el problema inicialmente planteado.

Sin duda, los objetivos marcados al principio del proyecto se han alcanzado exitosamente.

- ✓ Se ha desarrollado una aplicación para el rol *cliente* que permite ver la carta del restaurante, los detalles de los platos y también realizar el pedido.
- ✓ Se ha desarrollado la aplicación para el rol *manager* que recupera y muestra la información de los pedidos realizados en el restaurante.
- ✓ A lo que se añade que toda la información intercambiada entre aplicaciones se trata y almacena para poder ser consultada en cualquier momento.

Esto ha sido posible gracias a que desde un comienzo todos los requisitos planteados fueron cuidadosamente analizados para que quedaran integrados de la mejor forma posible en la aplicación.

Dichos requisitos cumplieron con las expectativas de los sub-objetivos también descritos al comienzo del proyecto.

- ✓ Las interfaces diseñadas son intuitivas y permiten a un usuario inexperto utilizar la aplicación fácilmente.
- ✓ El cliente puede modificar su pedido añadiendo platos en cualquier momento, sin necesidad de esperar a que un camarero quede disponible.
- ✓ El manager tiene acceso inmediato a la información enviada por los clientes. Además de ello, queda recogida en una única pantalla que puede tener continuamente a la vista y que le permite organizar la cocina y el restaurante rápidamente. Y por supuesto, tiene la opción de realizar búsquedas sobre pedidos terminados y operaciones sobre ambos tipos de pedidos.
- ✓ Las dos aplicaciones tienen una tolerancia a errores cometidos por el usuario muy alta. Ambas están preparadas para responder correctamente a cualquier error sin interferir en el funcionamiento normal de la misma, asegurando que el usuario no pueda realizar operaciones que no estén previstas.

A pesar de que estos fueron los objetivos marcados al principio, no han sido los únicos en alcanzarse, ya que a medida que se desarrollaba la aplicación iban surgiendo nuevas e interesantes funcionalidades que han mejorado la estructura, funcionalidad y rendimiento del sistema.

La planificación establecida al principio del proyecto se ha seguido meticulosamente para cumplir con los objetivos en los plazos fijados. Del mismo modo, los costes también se han ajustado para no superar las previsiones previstas.

Con todo ello se puede asegurar que los objetivos del producto se han conseguido satisfactoriamente en todos los casos.

## 5.1.2 Conclusiones Personales

Desde el primer momento he enfocado este proyecto como un reto personal.

Por un lado, tenía mucha curiosidad por programar en Android y ver qué podía hacer sabiendo que partía sin base ni experiencia en la plataforma. Por otro, llevaba tiempo queriendo realizar un proyecto que mejorara la organización y gestión de platos en la cocina de un restaurante y que además permitiera a los clientes ver y pedir de forma interactiva los platos.

La unión de ambas ideas se convirtió en mi proyecto de fin de carrera y ello me ha permitido aprender muchísimo, y sobre todo con muchas ganas, sobre diferentes tecnologías además de poner en práctica todo lo aprendido en la carrera en un proyecto más grande de lo acostumbrado en clase.

Ha sido un camino largo el que he recorrido desde el comienzo del proyecto. En él he encontrado innumerables obstáculos que, con más o menos esfuerzo, he conseguido superar. Hoy mirando atrás veo esos obstáculos insignificantes y me animan a continuar el camino, aun sabiendo que encontraré muchos más problemas en él. Pero todo este tiempo me ha enseñado que los problemas son divertidos y cuanto más difíciles son, más divertidos se vuelven.

Por ello puedo decir, que tanto la realización del proyecto como los años de carrera, han incrementado aun más mi interés por la ingeniería y por la resolución de cualquier tipo de problema. En definitiva, puede decirse que el resultado de todo este trabajo ha conseguido satisfacer más objetivos de los planteados en un principio, además de abrir nuevas perspectivas y caminos hacia el futuro.

## 5.2 Líneas Futuras

El proyecto presentado es una muy pequeña parte de la idea completa que se quería desarrollar en un principio, pero por complejidad y tiempo, no se han podido incluir todos los detalles.

La idea original era en parte distinta a la actual, y en un futuro próximo me gustaría hacer a la aplicación actual evolucionar a la aplicación inicialmente planteada.

A continuación se detallan las líneas futuras a implementar para conseguir que la aplicación sea lo más completa posible, de manera que se puedan gestionar tanto los pedidos, como otros aspectos de un restaurante:

- En lugar de tener dos aplicaciones distintas (una para el cliente y otra para el manager), se unificará toda la funcionalidad en una única aplicación y cada usuario accederá a un módulo de la misma dependiendo de su rol.
- En caso de caída del sistema, todos los módulos repondrán la última sesión establecida previa la caída. Con ello se evita pérdida y redundancia de información, además de la posibilidad de poder continuar sin problemas por donde la aplicación había quedado.
- La sincronización de las aplicaciones será automática.
- La configuración inicial de ambas aplicaciones se guardará en un archivo que el mismo programa leerá al iniciarse.

### **En el módulo del cliente de restaurante:**

- Existirá la opción de acceder a un mini-tutorial que enseñará a utilizar la aplicación a aquellos usuarios que lo deseen. Del mismo modo, existirá la opción de activar 'tips' o *pistas* indicando como hacer uso de ciertas funcionalidades mostradas en pantalla.
- No será un requisito obligatorio, pero el cliente tendrá la oportunidad de registrarse e iniciar sesión de forma privada para acceder a contenido personalizado, pudiendo:

- Consultar pedidos hechos anteriormente y poder volver a pedirlos sin tener que buscar por el menú.
  - Recibir ofertas del restaurante y aprovecharse de ventajas para los clientes habituales.
  - Personalizar la interfaz a su gusto.
- El precio total del pedido se podrá dividir y asignar a cada comensal, para que cada uno pague exclusivamente por lo que ha consumido.
- El cliente podrá ver los valores nutricionales (proteínas, grasas, azúcares, etc.) de los platos del menú, puestos en comparación con los valores diarios recomendados. Así mismo, también podrá establecer un límite de proteínas, grasas, kilocalorías y otros valores nutricionales para que el sistema genere un menú diario completo que no sobrepase dichos valores.
- Se podrá pagar directamente desde la aplicación, tanto si está registrado como si no. Para ello se hará uso del módulo de PayPal para Android.
- El cliente tendrá la opción de dejar su huella en un libro de firmas y también podrá dejar comentarios al restaurante a modo de sugerencias o críticas que ayuden a mejorar el servicio.
- Cada aplicación cliente tendrá una copia de la carta en la memoria interna del dispositivo y estará sincronizada y actualizada con la existente en base de datos. Las aplicaciones descargarán o actualizarán automáticamente la carta en el dispositivo cada vez que se produzca un cambio.
- Se introducirán más idiomas además del español.
- Los platos mostrados en los pedidos estarán ordenados de acuerdo con sus categorías y se permitirá al usuario indicar el orden de salida de cada plato.
- Existirá la opción de ver un menú del día y recomendaciones del chef.
- La interfaz de ingredientes a eliminar no mostrará solo aquellos que puedan eliminarse, sino todos los ingredientes del plato, haciéndose una diferenciación entre los que pueden eliminarse y los que no.



- Los clientes podrán consultar si los platos están ya realizándose en cocina, el tiempo aproximado de preparación y el tiempo hasta que lleguen a la mesa.

#### **En el módulo del manager:**

- Se pretende desarrollar una funcionalidad que permita llevar al día el inventario de los ingredientes necesarios en la cocina. Del mismo modo, se mantendrá un registro de la información de contacto de los proveedores, de forma que se puedan hacer pedidos en línea desde la misma aplicación rápidamente.
- El manager podrá consultar qué platos se están preparando en la cocina en ese momento, así como el tiempo restante de preparación aproximado.
- El manager podrá distinguir entre la cantidad de platos que han sido enviados al comedor, de manera que si no es posible enviar todos a la vez, pueda indicarlo para tenerlo en cuenta más adelante. Del mismo modo, será notificado si el cliente actualiza un pedido que ya se encuentra en cocina.
- Se podrán hacer análisis estadísticos a partir de cierta información recogida en la aplicación, como puede ser:
  - Número de personas por mesa que suelen visitar el restaurante a distintas horas del día.
  - Platos preferidos por los clientes, qué ingredientes eliminan, qué preferencias tienen en cuanto a cada plato, comentarios sobre la calidad, sabor, puntos a mejorar...
  - Número de veces que un cliente (registrado en la aplicación) vuelve al restaurante, con cuánta gente lo hace, etc.
  - Promedio de tiempo que se tarda en recibir un pedido desde cocina desde que este ha sido pedido, con respecto al número de comensales en la mesa. Esto lleva a que la aplicación *aprenda* y estime el tiempo que llevará a una mesa recibir sus pedidos con cada vez más precisión.

- Posibilidad de administrar la carta del restaurante desde una interfaz sencilla, pudiendo introducir nuevos platos, modificar o eliminar los existentes y ajustar otros datos importantes.
- El manager podrá comunicar cierta información a los clientes que estén esperando su pedido, como el tiempo que queda para recibir un plato, si no se pudiera cocinar un plato por falta de ingredientes, etc.
- Podrá eliminar platos de un pedido realizado, actualizándose a su vez en la interfaz del cliente.
- Los platos mostrados en los pedidos estarán ordenados de acuerdo con sus categorías y se mostrará el orden de salida indicado por el cliente.
- Se introducirá la opción de ver comentarios y sugerencias enviadas por los clientes para el restaurante en general.
- Se pretende mejorar la posibilidad de búsqueda de pedidos finalizados para poder introducir más criterios de búsqueda.
- Existirá conexión con la caja registradora y la impresora de tickets.

#### **Se creará un nuevo módulo para la cocina:**

Este módulo está pensado para proyectarse en la cocina y estar a la vista de los cocineros de forma continua, permitiéndoles saber en todo momento el estado de los pedidos, el tiempo que llevan en espera y otros detalles de interés.

Las ventajas y características que ofrece este nuevo módulo son:

- El jefe de cocina marca qué platos están preparados y pueden llevarse a la mesa. Esta funcionalidad se comunica directamente con el módulo del manager, que es quien indica a los camareros qué y dónde hay que servir. Hasta el momento, era el manager el que marcaba los platos que estaban listos, pero ahora su interfaz se actualiza automáticamente con la información enviada desde la cocina permitiendo mayor agilidad en el envío de platos al comedor.

- Cuando se comience a realizar un plato, el jefe de cocina marcará en su interfaz que el plato se encuentra en proceso. El manager recibirá esa información en su interfaz del manager y permitirá ejercer una mejor gestión del tiempo y recursos (en este caso, camareros) en el restaurante.

A rasgos generales, estas son algunas de las mejoras a introducir, aunque seguramente como ha ocurrido hasta el momento, por el camino vayan surgiendo ideas a desarrollar.

Es muy importante también contar con un buen feedback de la aplicación, de modo que se puedan limar asperezas, introducir cambios y mejoras sobre lo ya desarrollado, y contar con opiniones que lleven a pensar en nuevas funcionalidades para solventar problemas que no se han tenido en cuenta aún.



# Glosario

---

ADT	<i>Android Development Tool</i>
API	<i>Application Programming Interface</i>
APK	<i>Android Package</i>
BBDD	<i>Base de Datos</i>
CRUD	<i>Create, Read, Update and Delete</i>
DAO	<i>Data Access Object</i>
Endpoint	<i>Extremo final de un elemento. En el caso de REST y SOAP, se refiere a los recursos que ubican la representación de los objetos de información</i>

Firewall	<i>Dispositivo software de seguridad que controla el tráfico de entrada y salida a la red de acuerdo con una política de control de acceso</i>
Framework	<i>Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado</i>
GB	<i>Gigabyte</i>
GHz	<i>Gigahercio</i>
GPS	<i>Global Positioning System</i>
HTTP	<i>HyperText Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
IVA	<i>Impuesto de Valor Añadido</i>
JAX-RS	<i>Java API for XML RESTful Services</i>
JSON	<i>JavaScript Object Notation</i>
JDK	<i>Java Development Kit</i>
Mbps	<i>Megabits por segundo</i>
MVC	<i>Modelo-Vista-Controlador</i>
Plugin	<i>Componente software que añade capacidades específicas sobre una aplicación software de mayor envergadura</i>
POJO	<i>Plain Old Java Object</i>
Proxy	<i>Programa o dispositivo que realiza una acción en representación de otro</i>
RAM	<i>Random Access Memory</i>
RPC	<i>Remote Procedure Call</i>

SDK	<i>Software Development Kit</i>
SGBD	<i>Sistema Gestor de Base de Datos</i>
SOAP	<i>Simple Object Access Protocol</i>
SQL	<i>Structured Query Language</i>
TPV	<i>Terminal Punto de Venta</i>
URL	<i>Uniform Resource Locator</i>
Wi-Fi	<i>Wireless Fidelity</i>
WSDL	<i>Web Services Description Language</i>
XML	<i>EXtensible Markup Language</i>





# Anexo A

## Manual de Usuario Manager

---

Este anexo tiene como objetivo mostrar en detalle cómo desenvolverse con la aplicación destinada al manager, así como a resolver los posibles problemas que puedan surgir durante su uso. Para ello, se establecerán una serie de pasos a seguir para aprender a navegar por la interfaz a modo de tutorial.

## **Paso 1. Iniciar la aplicación.**

Tras iniciar la aplicación se muestra el logotipo de la aplicación y se espera hasta que el usuario toca la pantalla. Esa acción le lleva a la siguiente pantalla, que corresponde a la interfaz de *Pedidos en curso*.

## **Paso 2. Trabajar con pedidos en curso.**

Nos encontramos en la pantalla de pedidos activos. En ella pueden darse diferentes situaciones que veremos en detalle.

### **1. No existen pedidos en curso**

Si no hay ningún pedido actualmente en curso en el restaurante, esta pantalla aparecerá vacía, mostrando únicamente un mensaje advirtiendo de tal hecho y un botón de actualizar.

### **2. Queremos actualizar el contenido**

Si queremos refrescar la pantalla para consultar la información actualizada, pulsaremos siempre el botón de actualizar. Este botón aparece tanto si hay como si no hay pedidos en curso.

### **3. Consultar información de pedidos en curso.**

Cuando en la lista del lado izquierdo aparezca al menos una mesa, procedemos a consultar la información de un pedido. Para ello, seleccionamos de la lista de mesas uno cualquiera y nos fijamos en los detalles que aparecen en el lado derecho, que corresponden a:

- Número de mesa.
- Listado de platos pedidos y comentarios.
- Indicador de plato hecho y servido en la mesa.
- Precio total a pagar.
- Botones de actualizar interfaz e imprimir cuenta.

#### **4. Marcar platos que han sido preparados**

Cada vez que un plato pedido se envíe a la mesa, ha de marcarse el checkbox junto a dicho plato en el pedido correspondiente. De este modo, se lleva el orden de los platos que han sido servidos y los que aún están por servir.

#### **5. Comprobar y actuar si el cliente pide la cuenta**

Cada vez que un cliente pida la cuenta, el manager será notificado con un icono verde con el símbolo del euro en el listado de mesas, junto al nombre de la mesa que la pidió.

En este punto, el manager pulsaría el botón *Imprimir Cuenta* para poder llevar el ticket a la mesa y podrá ver en pantalla un mensaje indicando que el ticket se está imprimiendo. Este botón estará siempre visible por si fuera necesario en cualquier momento.

Cuando el cliente pide la cuenta aparece el botón *Marcar Pagado* junto a los detalles de la mesa. Este botón deberá ser pulsado una vez el cliente haya pagado el importe del pedido para dejar constancia de ello. Al hacerlo, aparecerá en los detalles un icono con el texto *Pagado* en su interior.

#### **6. Finalizar un pedido**

Para poder finalizar un pedido existen dos requisitos a cumplir:

- Que todos los platos del pedido a finalizar estén marcados.
- Que el cliente haya pedido la cuenta y se haya marcado como pagado.

Si ambas condiciones se cumplen, aparecerá un botón con el texto *Terminar* que tras pulsarlo, archivará el pedido para poder ser consultado desde el historial de pedidos, desapareciendo de la interfaz de pedidos en curso.

#### **7. Consultar el historial de pedidos**

Nos dirigimos a la parte superior de la pantalla, donde se encuentran las pestañas de Pedidos en curso e Historial de pedidos. Acceder al historial de pedidos es tan fácil como pulsar la pestaña con dicho nombre.

## Paso 3. Trabajar con el historial de pedidos

Tras seleccionar la pestaña de Historial de pedidos, nos encontramos en la interfaz que nos permite realizar búsquedas de pedidos que han sido finalizados. Pueden darse distintas situaciones que veremos ahora en detalle.

### 1. No existen pedidos en el historial

En caso de que el historial no tenga ningún pedido, se mostrará el contenido de la interfaz vacía. Si queremos que aparezca algún pedido en el historial, se ha de finalizar desde la interfaz de pedidos en curso.

### 2. Consultar información de un pedido terminado

Cuando en la lista del lateral izquierdo aparezca al menos un pedido, procedemos a consultar su información. Para ello, seleccionamos de la lista uno cualquiera y nos fijamos en los detalles que aparecen en el lado derecho, que corresponden a:

- Número de mesa desde donde se realizó el pedido.
- Listado de platos pedidos y comentarios.
- Fecha y hora en que se realizó el pedido.
- Precio total del pedido.
- Indicador de si el pedido ha sido pagado (en caso de no aparecer, indica que el pedido no ha sido pagado).
- Botón de imprimir cuenta.

### 3. Establecer filtro de búsqueda por Número de mesa

Este filtro devuelve todos los pedidos finalizados que hayan sido servidos en la mesa especificada.

Para introducir el filtro, se pulsa el cuadro de texto que está a la derecha del título *Nº Mesa* y se especifica el valor exacto de la mesa cuyos pedidos se quieren recuperar. Solo es posible especificar una mesa en cada búsqueda y no se permiten nombres parciales.

#### **4. Establecer filtro de búsqueda por Precio**

Este filtro devuelve todos los pedidos cuyo precio sea mayor o igual que el precio mínimo y/o menor o igual que el precio máximo especificado.

Si se quiere introducir un valor exacto, ambos filtros deberán tener el mismo valor.

Para introducir el filtro, se pulsa el cuadro de texto que está a la derecha del título *Precio*, siendo el más cercano al título el precio mínimo y el más lejano el precio máximo. Este campo admite valores con dos dígitos decimales.

#### **5. Establecer filtro de búsqueda por Plato**

Este filtro devuelve todos los pedidos que contengan el plato especificado.

No es necesario introducir el nombre completo del plato. El sistema siempre hará una búsqueda de resultados en cuyos pedidos existan platos cuyo nombre contenga la cadena de caracteres introducida.

Para introducir el filtro, se pulsa el cuadro de texto que está a la derecha del título *Plato* y se introduce el nombre, o parte del nombre, del plato deseado.

#### **6. Establecer filtro de búsqueda por ID de pedido**

Si se especifica el ID completo, se devolverá un único resultado, ya que el identificador es unívoco.

Si se introduce parte de la cadena, se devolverán todos los resultados encontrados en cuyo identificador exista la cadena especificada.

Para introducir el filtro, se pulsa el cuadro de texto que está a la derecha del título *Id Pedido* y se introduce el ID, o parte del ID, del pedido deseado.

Esta búsqueda, a pesar de no parecer útil a simple vista, es la única que permite a un usuario hacer búsquedas que devuelvan un elemento concreto en lugar de una serie de ellos.

## **7. Establecer filtro de búsqueda por Fecha inicial y Fecha final**

Este filtro devuelve los pedidos cuyas fechas sean mayores o iguales, o menores o iguales (respectivamente) a las establecidas.

No es necesario introducir ambas a la vez, a no ser que se quiera recuperar únicamente los pedidos correspondientes a un día exacto.

Para introducir la fecha, pulsaremos en el espacio junto al texto e inmediatamente se abrirá un selector de fecha que permitirá elegir la fecha deseada fácilmente. Una vez seleccionada la correcta, pulsaremos aceptar.

Si deseamos eliminar los filtros, pulsaremos el aspa roja junto a la fecha que queramos borrar.

## **8. Buscar con los filtros establecidos**

Después de establecer todos los filtros que se deseen, se pulsa el botón *Buscar* ubicado en el lateral derecho.

Si existen coincidencias, se rellenará el listado de pedidos finalizados con aquellos que cumplan los criterios de búsqueda.

Si no existen coincidencias, el usuario es notificado mediante un mensaje emergente y el resultado de las búsquedas aparecerá vacío.

## **9. Recuperar todos los pedidos finalizados**

Al navegar desde la interfaz de pedidos en curso al historial, se muestran el listado completo de pedidos finalizados.

Si se han introducido filtros y se desea recuperar todos los pedidos finalizados, deberán eliminarse todos los filtros introducidos y pulsar el botón buscar.

## **10. Imprimir ticket**

Para imprimir el ticket de un pedido finalizado, se debe presionar el botón *Imprimir Cuenta* después de haber seleccionado el pedido que desea imprimirse. Aparecerá un mensaje indicando que el ticket está imprimiéndose, que se cerrará en pocos segundos. Puede imprimirse tantas veces como se necesite.

# Anexo B

## Manual de Usuario Cliente

---

Este anexo tiene como objetivo mostrar en detalle cómo desenvolverse con la aplicación destinada al cliente del restaurante, así como a resolver los posibles problemas que puedan surgir durante su uso. Para ello, se establecerán una serie de pasos a seguir para aprender a navegar por la interfaz a modo de tutorial.

## Paso 1. Bienvenida

La primera pantalla que encuentra el cliente es la pantalla de bienvenida. En ella, además de recibirle, se le pide que introduzca el número de comensales que serán. Para ello, se debe seleccionar el número correspondiente en el selector, bien se:

- Utilizando las flechas de dirección arriba y abajo.
- O manteniendo pulsado y arrastrando con el dedo hasta llegar al número correcto.

Una vez seleccionado se debe pulsar el botón continuar, que llevará al usuario a la interfaz de la carta del restaurante.

## Paso 2. Consultar la carta

Después de la bienvenida, al cliente se le presenta la carta del restaurante.

Dado que se ha querido mantener toda la información recogida en el menor número de interfaces, en esta pantalla se pueden realizar muchas acciones y darse distintas situaciones que veremos ahora en detalle.

### 1. Selección de categorías de platos

Como en la carta de cualquier restaurante, los platos están agrupados dependiendo de la categoría a la que pertenezcan, pudiendo así tener categorías de ensaladas, pasta, carne, entrantes, bebidas, postres, etc.

En la aplicación sucede lo mismo y para consultar los diferentes platos y bebidas que se ofrecen en la carta, se ha de seleccionar una de las categorías existentes en la parte superior de la pantalla.

### 2. Selección de plato

Después de seleccionar una categoría, en la parte inferior izquierda se muestra el listado de todos los platos asociados a esa categoría y que pueden pedirse en el restaurante.

Para seleccionar uno de los platos y acceder a su contenido, se ha de tocar cualquiera de los existentes en la lista.



### 3. Consulta de información sobre los platos

Ya tenemos un plato seleccionado y ahora queremos ver qué tipo de información podemos obtener de él.

Dirigimos la mirada hacia la parte derecha de la pantalla, donde veremos una foto del plato, el nombre, una descripción, el precio y, en caso de que el plato tenga algún comentario hecho por clientes en visitas previas, también habrá una valoración representada con entre 1 y 5 estrellas y un comentario visible sobre dicho plato.

Además de ello, existen dos pestañas verticales, estando la primera de ellas seleccionada en esta vista ya que muestra la información del plato. Si se selecciona la segunda pestaña, la parte donde se muestran los comentarios y la descripción desaparece, dando paso a las opciones del plato.

### 4. Consultar de opciones del plato

Después de seleccionar la pestaña *Opciones*, se muestran en la interfaz de la carta del restaurante uno o dos botones, dependiendo de la situación.

#### ✓ Botón de Enviar comentario a cocina

El primero de los botones tiene la funcionalidad de enviar un comentario a la cocina, lo que permite al usuario dar ciertas instrucciones al cocinero antes de preparar dicho plato (por ejemplo, si se quiere la carne poco hecha, con poca sal, etc.).

Este botón estará siempre disponible y al presionarlo, aparecerá bajo los botones un cuadro de texto y un teclado software que permitirá al usuario escribir las órdenes de preparación.

Se puede introducir el comentario tanto antes como después de eliminar un ingrediente, pero este se borrará si se selecciona otro plato, por lo que antes de hacerlo es importante añadir el plato con comentario al pedido.

### ✓ **Botón de Eliminar ingrediente**

El segundo botón aparecerá únicamente si el plato seleccionado tiene ingredientes que puedan ser eliminados.

Al pulsarlo, se abre una nueva interfaz con el conjunto de iconos y nombres de todos aquellos ingredientes incluidos en el plato que se puedan eliminar.

Para eliminar un ingrediente, deberá seleccionarse hasta quedar marcado en gris con un aspa roja encima. Una vez seleccionados todos los ingredientes que quieren eliminarse, se pulsará el botón *Hecho* para volver a la carta.

## **5. Consulta de comentarios del plato**

En caso de que un plato haya sido comentado y valorado en más de una ocasión, se pueden ver todos los comentarios tocando una vez sobre el texto. A medida que se toque este espacio, se irán mostrando nuevos comentarios hasta llegar al final, y en ese momento se comenzará de nuevo por el primero.

La valoración de estrellas que se muestra es el promedio de todas las valoraciones hechas sobre el plato.

## **6. Añadir plato al pedido**

Si se desea añadir un plato al pedido, se ha de pulsar el botón de la parte inferior de la pantalla con el texto *Añadir*. Se añadirán tantos platos iguales como veces se pulse esté botón.

## **7. Ver el pedido**

Bajo el listado de platos se encuentra el botón *Ver Pedido*. Al pulsarlo, se deja atrás la interfaz de la carta para pasar a ver una nueva pantalla que muestra el pedido de esta sesión.

## Paso 3. Consultar el pedido

Para comodidad del usuario a la hora de consultar el pedido, el sistema diferencia entre los platos que han sido añadidos a la lista pero aun están pendientes de confirmación para enviar a la cocina, y los platos que ya han sido revisados y enviados para su preparación al cocinero.

Por ello, encontramos dos pestañas que diferencian entre el pedido en curso y el pedido en cocina.

### 1. Vista de Pedido en curso

Después de pulsar el botón *Ver Pedido* en la carta, se muestra la interfaz de pedido en curso, donde se dispone de diferentes opciones.

#### ✓ Consultar la información del pedido en curso

Esta interfaz muestra el listado de todos los platos que se han añadido desde la carta y aun no han sido enviados a la cocina, junto con la cantidad de veces que se quiere dicho plato, además del precio por plato y el precio total de todos ellos.

#### ✓ Incrementar o reducir la cantidad de cada plato

El número que indica la cantidad de cada plato pedida tiene a su izquierda un botón con el símbolo + y a su izquierda un botón con el símbolo -. Al pulsarlos, la cantidad de cada plato pedido se incrementa o reduce al gusto del cliente.

#### ✓ Eliminar plato

Si se desea eliminar un plato, deberá utilizarse el botón de reducir la cantidad de un plato hasta que el valor alcance la cifra 0.

#### ✓ Consultar comentario para cocina

Si se ha escrito un comentario para el cocinero en alguno de los platos añadidos, este puede volver a consultarse desde el listado de platos de esta interfaz. Junto al plato que tenga un comentario aparece un icono que, al pulsarlo, muestra una ventana emergente con el comentario introducido.

✓ **Enviar el pedido a la cocina**

Cuando el cliente quiera confirmar el pedido y que este se envíe a la cocina para comenzar su preparación, deberá pulsar el botón de la parte inferior con el texto *Pedir*. Después de enviar un pedido no se puede modificar, pero se pueden seguir pidiendo platos adicionales hasta que se quiera.

✓ **Añadir más platos**

Si aun nos encontramos en la vista de pedidos en curso, o si ya hemos visitado la de pedidos en cocina, podemos seguir añadiendo platos al pedido dirigiéndonos para ello a la carta de nuevo.

Para volver a la carta, utilizaremos el botón de la parte inferior con el texto *Ver Carta* y seguiremos añadiendo los platos como hicimos con anterioridad. Todos los platos añadidos se sumarán a los que ya estaban añadidos anteriormente y se necesitará volver a confirmar el envío a la cocina para su preparación.

✓ **Consultar los pedidos enviados a la cocina**

Si se ha enviado el pedido a la cocina, aparecerá junto a la pestaña de *Pedido en curso* una segunda pestaña que, tras pulsarla, mostrará la interfaz con la información del pedido enviado a la cocina.

## **2. Vista de Pedido en cocina**

Después de pulsar el botón *Pedido realizado*, se muestra la interfaz del pedido que ha sido confirmado y enviado a la cocina durante la sesión actual. Esta pestaña solo aparece si se ha enviado un pedido, en otro caso, permanecerá escondida e inactiva.

En ella se ofrecen diferentes opciones.

✓ **Consultar la información del pedido realizado**

Esta interfaz muestra el listado de todos los platos que se han enviado a la cocina, junto con la cantidad de veces que se quiere dicho plato, además del precio por plato y el precio total de la suma de todos ellos.

### ✓ **Valorar un plato pedido en el restaurante**

El sistema permite al usuario valorar y dejar un comentario visible para otros usuarios sobre un plato que haya degustado durante su visita.

Para ello, debe pulsar el icono existente junto al precio de cada plato en esta interfaz. Al hacerlo, se muestra en el lateral derecho el nombre del plato que se ha seleccionado, un cuadro de texto sobre el que el cliente introducirá su comentario y un valorador que permiten dar al plato una puntuación de 1 a 5 estrellas.

Para valorar el plato, tiene que introducirse tanto un comentario como un número de estrellas mayor que cero y a continuación pulsar el botón *Aceptar*. A partir de ese momento, cualquier cliente nuevo que utilice la aplicación podrá ver el comentario junto al plato correspondiente.

Pulsando el botón *Borrar*, se borrará el texto introducido sin llegar a enviarse.

### ✓ **Pedir la cuenta**

Un usuario puede pedir la cuenta en cualquier momento, pero después de hacerlo no podrá seguir realizando pedidos en la misma sesión.

Para pedir la cuenta ha de pulsar el botón *Pagar*. Aparecerá un mensaje informando al cliente de que no podrá realizar más pedidos en caso de pedir la cuenta y dos opciones: una que confirma que desea pedir la cuenta y otro que cancela la opción.

Cancelando la opción se puede continuar utilizando el sistema normalmente.

Confirmando que se quiere recibir la cuenta, la sesión termina y se agradece al cliente su visita.



# Anexo C

## Implantación

---

Este anexo tiene como finalidad definir los requisitos técnicos necesarios a la hora de la implantación del sistema en un restaurante, así como describir los pasos a realizar para la instalación y configuración de los componentes necesarios para el correcto funcionamiento de la aplicación.

## 1. Requisitos técnicos

Antes de comenzar la implantación, es necesario disponer de ciertos elementos esenciales en el restaurante, que, de no tenerse, habrán de adquirirse o contratarse. Estos son:

- **Red Wi-Fi:** mínimo a 54 Mbps.
- **Dispositivos tablet:** de las características acordadas en el capítulo 2.
  - 1 dispositivo para la aplicación mánager.
  - Tantos dispositivos como mesas haya en el restaurante, para la aplicación cliente.
- **Ordenador:** portátil o de sobremesa, indiferentemente. Como mínimo ha de reunir las características siguientes:
  - Dual Core a 2.4 GHz con 4GB RAM y 500 GB Disco duro.
  - Sistema operativo: Windows XP professional.

La conexión a internet mediante Wi-Fi será necesaria para realizar la comunicación cliente-servidor. Debe tratarse de una conexión rápida para agilizar el proceso de intercambio de información, por lo que se requiere una conexión Wi-Fi de 54 Mbps.

Se establecerá el servidor en el propio restaurante, permitiendo mayor agilidad en los accesos al estar en una red local. Para ello será necesario contar con un ordenador, que puede ser tanto portátil como de sobremesa, en el que también se integra la base de datos.



## 2. Configuración del sistema

Para facilitar la implantación del sistema, se utilizará una serie de ficheros de configuración que se encargarán de instalar y establecer todos los parámetros necesarios automáticamente.

Los ficheros estarán distribuidos en una jerarquía de directorios como la siguiente.

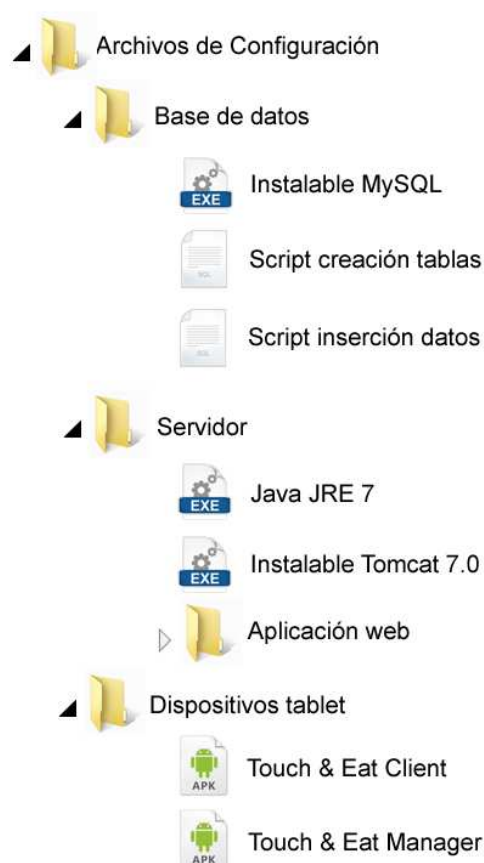


Figura 60 – Jerarquía de directorios de los archivos de configuración

Será necesario seguir los siguientes pasos para completar el proceso de instalación.

### 1. Configuración de Base de Datos

La base de datos se configurará sobre el ordenador portátil o de sobremesa especificado en los requisitos técnicos. Para comenzar, debe copiarse el contenido del directorio *Base de Datos* en él.

Una vez copiado, se inicia la instalación de la base de datos ejecutando el archivo *Instalable MySQL* y se siguen los pasos indicados en las sucesivas pantallas hasta su completa instalación.

Tras terminar la instalación, se ejecuta el archivo *Script creación tablas* desde el terminal del sistema. Este archivo contiene las queries en lenguaje SQL necesarias para crear la base de datos y las tablas especificadas en el capítulo de Diseño de la Base de Datos.

Teniendo la base de datos y tablas creadas, se ejecuta del mismo modo el *Script inserción datos*, que contiene las sentencias SQL para la inserción de datos en las tablas que lo requieren.

## **2. Configuración del servidor**

El servidor web se establece en el ordenador portátil o de sobremesa especificado en los requisitos técnicos. Para comenzar, debe copiarse el contenido del directorio *Servidor* en él.

Para instalar el servidor Tomcat debe disponerse de una distribución de Java en el sistema, por lo que se instalará previamente.

Para ello, se ejecuta el archivo Java JRE 7 y se siguen los pasos indicados hasta su completa instalación.

Tras ello, se inicia la instalación del servidor ejecutando el archivo *Instalable Tomcat 7.0* y se siguen los pasos para completar su instalación. Deberá marcarse la opción de *auto-startup* para permitir que el servicio se inicie automáticamente al iniciar la sesión de Windows.

Una vez instalado el servidor, se moverá la carpeta *Aplicación web* dentro del directorio `$DIRECTORIO_BASE/webapps/`, donde `$DIRECTORIO_BASE` apunta a la dirección donde Tomcat ha sido instalado. El directorio *Aplicación web* contiene el código creado para la comunicación entre aplicaciones y base de datos.

Tras reiniciar Tomcat, el servidor está listo para enviar y recibir peticiones.

### 3. Configuración de los dispositivos tablet

Deberá haber un dispositivo tablet para la aplicación manager y tantos dispositivos tablet como mesas existan en el restaurante para las aplicaciones cliente.

En la tablet destinada al manager debe copiarse el archivo apk *Touch & Eat Manager*, mientras que en cada una de las tablets destinadas a los clientes debe copiarse el archivo *Touch & Eat Client*.

Una vez copiado el fichero apk correspondiente a cada rol, deberá ser localizada su ubicación en cada dispositivo tablet a través de un programa gestor de ficheros. Una vez localizado, se seleccionará para comenzar su instalación.

Tras realizar los mismos pasos en todos los dispositivos, se crea un icono en el menú de aplicaciones que debe pulsarse para comenzar a utilizar la aplicación.

Al iniciar la aplicación es necesario configurar la conexión al servidor. Para ello, debe introducirse la dirección al servidor con formato <IP>:<Puerto>. Para obtener la IP, en el ordenador donde se encuentra el servidor debe abrirse un terminal y hacer una llamada a *ipconfig*, que devolverá la dirección IP entre otros detalles. El puerto será 8080, dado que así está configurado el servidor web.

En el caso del cliente, deberá además indicarse el número de mesa correspondiente a la tablet.

Una vez establecidos estos parámetros, pulsando el botón *Continuar* quedará configurada la aplicación durante esa sesión, pudiendo comenzar a utilizarse con normalidad.



# Bibliografía

---

- [1] Murphy, Mark. *Beginning Android 3*. Apress 2010.
- [2] Leonard Richardson, Sam Ruby. *RESTFull Web Services*. O'Reilly 2007.
- [3] W. Frank Ableson, Charlie Collins, Robi Sen. *Unlocking Android. A Developer's Guide*. Manning Publications Co 2009.
- [4] Tecnologías de la información y la comunicación en la hostelería. Octubre 2012.  
[http://innovacion.gijon.es/multimedia\\_objects/download?object\\_type=document&object\\_id=948](http://innovacion.gijon.es/multimedia_objects/download?object_type=document&object_id=948)
- [5] Definición y características de los dispositivos móviles. Octubre 2012.  
<http://revista.seguridad.unam.mx/numero-07/dispositivos-m%C3%B3viles>
- [6] Infografía. El crecimiento de la telefonía móvil. Octubre 2012.  
<http://www.ticbeat.com/sim/crecimiento-telefonía-movil-infografía/>

- [7] Disposición de telefonía móvil a nivel mundial. Octubre 2012.  
<http://blog.hostalia.com/se-calcula-que-el-92-de-la-poblacion-mundial-dispone-de-telefono-movil/>
- [8] Estimación de la población mundial. Enero 2012.  
[http://es.wikipedia.org/wiki/Poblaci%C3%B3n\\_mundial](http://es.wikipedia.org/wiki/Poblaci%C3%B3n_mundial)
- [9] Presentación de estudio “Mobile Life” por TNS. Octubre 2012.  
<http://www.slideshare.net/TNSspain/presentacin-del-estudio-mobile-life>
- [10] Google Play consigue 25 mil millones de descargas. Octubre 2012.  
<http://officialandroid.blogspot.com.es/2012/09/google-play-hits-25-billion-downloads.html>
- [11] Google Play festeja 25 mil millones de descargas con descuentos. Octubre 2012.  
<http://www.softoy.com/google-play-festeja-25-mil-millones-descargas-descuentos.html>
- [12] Características del Sistema Operativo Android. Octubre 2012.  
[http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [13] Características del Sistema Operativo iOS. Octubre 2012.  
<http://en.wikipedia.org/wiki/iOS>
- [14] Características del Sistema Operativo Symbian. Octubre 2012.  
<http://en.wikipedia.org/wiki/Symbian>
- [15] Características del Sistema Operativo Blackberry. Octubre 2012.  
[http://en.wikipedia.org/wiki/BlackBerry\\_OS](http://en.wikipedia.org/wiki/BlackBerry_OS)
- [16] Aplicaciones disponibles en App World. Octubre 2012.  
<http://crackberry.com/99500-apps-available-blackberry-app-world-25-percent-which-were-built-blackberry-playbook>
- [17] Análisis de Windows Phone. Octubre 2012.  
[http://www.webopedia.com/TERM/W/windows\\_phone.html](http://www.webopedia.com/TERM/W/windows_phone.html)
- [18] Aplicaciones disponibles en Windows Phone Store. Octubre 2012.  
[http://news.cnet.com/8301-10805\\_3-57542096-75/microsoft-windows-phone-store-up-to-120000-apps/](http://news.cnet.com/8301-10805_3-57542096-75/microsoft-windows-phone-store-up-to-120000-apps/)
- [19] Comparación de dispositivos con Symbian. Octubre 2012.  
[http://en.wikipedia.org/wiki/Comparison\\_of\\_Symbian\\_devices](http://en.wikipedia.org/wiki/Comparison_of_Symbian_devices)

[20] Infografía: Samsung Galaxy Tab 10.1 vs. Apple iPad 2 vs. HP TouchPad vs. LG G-Slate. Octubre 2012.

<http://skattertech.com/2011/03/infographic-samsung-galaxy-tab-10-1-vs-apple-ipad-2-vs-hp-touchpad-vs-lg-g-slate/>

[21] Comparación de características y precios de iPad. Octubre 2012.

<http://www.apple.com/es/ipad/compare/>

[22] Comparación de dispositivos que corren bajo Android. Octubre 2012.

[http://en.wikipedia.org/wiki/Comparison\\_of\\_Android\\_devices#Tablet\\_computers](http://en.wikipedia.org/wiki/Comparison_of_Android_devices#Tablet_computers)

[23] Especificaciones Samsung Galaxy Tab 10.1. Octubre 2012.

<http://www.samsung.com/global/microsite/galaxytab/10.1/spec.html>

[24] Especificaciones Motorola Xoom. Octubre 2012.

[http://www.motorola.com/us/consumers/MOTOROLA-XOOM/72804,en\\_US,pd.html?selectedTab=tab-2&cgid=tablets#tab](http://www.motorola.com/us/consumers/MOTOROLA-XOOM/72804,en_US,pd.html?selectedTab=tab-2&cgid=tablets#tab)

[25] Especificaciones Asus Eee Pad Transformer. Octubre 2012.

[http://www.asus.es/Eee/Eee\\_Pad/Eee\\_Pad\\_Transformer\\_TF101/#specifications](http://www.asus.es/Eee/Eee_Pad/Eee_Pad_Transformer_TF101/#specifications)

[26] Evolución del número de aplicaciones en Google Play. Octubre 2012.

[http://en.wikipedia.org/wiki/Google\\_Play#Applications](http://en.wikipedia.org/wiki/Google_Play#Applications)

[27] Evolución del número de aplicaciones en Apple Store. Octubre 2012.

[http://en.wikipedia.org/wiki/App\\_Store\\_\(iOS\)#Number\\_of\\_launched\\_applications](http://en.wikipedia.org/wiki/App_Store_(iOS)#Number_of_launched_applications)

[28] Dispositivos sobre los que se puede ejecutar *OrderingApps*. Noviembre 2012.

<http://www.orderingapps.com/devices/>

[29] Página principal de la aplicación *OrderingApps*. Noviembre 2012.

<http://www.orderingapps.com>

[30] Página principal de la aplicación *Snapfinger*. Noviembre 2012.

<http://www.snapfinger.com/Info/Mobile>

[31] Vídeo de demostración sobre cómo utilizar *Snapfinger*. Noviembre 2012.

<http://www.youtube.com/watch?v=IY4pCQX-6VU>

[32] Página principal de la aplicación *Food2U*. Noviembre 2012.

<http://www.food2u.es/>

- [33] Restaurantes a los que se pueden hacer pedidos desde *Food2U*. Noviembre 2012.  
<http://www.food2u.es/cities.php>
- [34] Detalles y descarga de QlikBar. Noviembre 2012.  
<https://play.google.com/store/apps/details?id=com.glikbar&hl=en>
- [35] Qué es QlikBar. Noviembre 2012.  
<http://www.eureka-startups.com/blog/2012/06/22/glikbar-relacion-hosteleros-clientes/>
- [36] Dónde se puede usar QlikBar. Noviembre 2012.  
<http://www.glikbar.com/>
- [37] Descarga de Eclipse IDE for Java EE Developers. Diciembre 2012.  
<http://www.eclipse.org/downloads/packages/release/indigo/sr2>
- [38] Descarga e instalación del SDK de Android. Diciembre 2012.  
<http://developer.android.com/sdk/installing/adding-packages.html>
- [39] Características del Plugin ADT Eclipse. Diciembre 2012.  
<http://developer.android.com/tools/index.html>
- [40] Análisis de REST vs SOAP. Diciembre 2012.  
<http://geeknizer.com/rest-vs-soap-using-http-choosing-the-right-webservice-protocol/>
- [41] Pros y contras de la arquitectura RESTful. Diciembre 2012.  
<http://programmers.stackexchange.com/questions/23386/pros-and-cons-of-restful-architecture>
- [42] Tutorial de REST. Diciembre 2012.  
<http://rest.elkstein.org/>
- [43] Tutorial de REST con Java JAX-RS utilizando Jersey. Diciembre 2012.  
<http://www.vogella.com/articles/REST/article.html>
- [44] Jersey como implementación de JAX-RS. Diciembre 2012.  
<http://danipenaperez.blogspot.com.es/2012/01/jersey-la-implementacion-de-sun-para.html>
- [45] Tutorial de Jersey. Diciembre 2012.  
<http://docs.oracle.com/javaee/6/tutorial/doc/gilik.html>
- [46] Tutorial JAXB. Diciembre 2012.  
<http://www.vogella.com/articles/JAXB/article.html>
- [47] Comparación de implementaciones JAX-RS. Diciembre 2012.  
<http://www.infoq.com/news/2008/10/jaxrs-comparison>



- [48] Características y descarga de la librería Google GSON. Diciembre 2012.  
<http://code.google.com/p/google-gson/>
- [49] Características de Tomcat. Diciembre 2012.  
<http://tomcat.apache.org/>
- [50] Características de SQL Server. Diciembre 2012.  
<http://www.microsoft.com/sqlserver/en/us/product-info/overview-capabilities.aspx>
- [51] Características de MySQL. Diciembre 2012.  
[http://dev.mysql.com/usingmysql/get\\_started.html](http://dev.mysql.com/usingmysql/get_started.html)
- [52] Características de Oracle Database. Diciembre 2012.  
[http://en.wikipedia.org/wiki/Oracle\\_Database](http://en.wikipedia.org/wiki/Oracle_Database)
- [53] Análisis de distintos Servidores de aplicaciones. Diciembre 2012.  
<http://www.icm.es/tecnologias/servidores-de-aplicaciones/>
- [54] Salario medio de un programador. Diciembre 2012.  
<http://www.tufuncion.com/trabajo-programador>
- [55] Herramienta cálculo de salarios medios *InfoJobs Trends Salarios*. Diciembre 2012.  
<http://salarios.infojobs.net/>